Université de Strasbourg

UNIVERSITÉ DE STRASBOURG

École doctorale Mathématiques, sciences de l'information et de l'ingénieur | ED 269 Université de Strasbourg

ÉCOLE DOCTORALE MATHEMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGENIEUR – ED269

[ICube laboratory]

THÈSE présentée par :

[Kaifeng ZOU]

soutenue le : **28/09/2023**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : SIAR (Signal, Image, Automatique, Robotique)

Advancements in Generative Models: Enhancing Interpretability and Control of Complex Data through Disentanglement and Conditional Generation

THÈSE dirigée par :

Sylvain Faisan	Maître de conférences, Université de Strasbourg, France
Fabrice Heitz	Professeur des Universités, Université de Strasbourg, France
Sébastien Valette	Chargé de recherches, CNRS, INSA-Lyon, France

RAPPORTEURS :

Marco Lorenzi	Chargé de recherches, Inria, Université Côte d'Azur, France
Su Ruan	Professeur des Universités, Université de Rouen Normandie, France

AUTRES MEMBRES DU JURY :

Pierre Charbonnier	Directeur de recherches, Cerema Strasbourg, France

Résumé en français

Cette thèse a été rédigée en anglais. Voici un résumé détaillé de la thèse en français.

Contexte

L'intelligence artificielle (IA) est devenue un thème récurrent dans les films de science-fiction contemporains, comme on peut le voir avec des personnages tels que Joi dans *Blade Runner 2049* et Jarvis dans *Iron Man*. Ces représentations dépeignent souvent des systèmes d'IA dotés de capacités comparables à celles des humains.

Aujourd'hui, l'apprentissage profond permet aux systèmes d'IA d'accomplir des tâches (générer du texte, des images...) qui étaient autrefois réservées aux humains. La clé de cette technologie réside dans l'IA générative. Ces modèles capturent la distribution sous-jacente des données $x \sim p_D(x)$ et utilisent ces connaissances pour générer de nouveaux échantillons $\hat{x} \sim p_\theta(x)$, où θ représente les paramètres du modèle.

Les modèles génératifs, tels que les auto-encodeurs variationnels, les réseaux antagonistes génératifs, les réseaux de flot génératifs et les modèles de diffusion, ont montré un potentiel significatif dans divers domaines, notamment la génération d'images, la synthèse de la parole et le traitement du langage naturel.

Les modèles génératifs ont connu de nombreux progrès ces dernières années. Ces avancées ont été motivées par plusieurs facteurs, notamment la disponibilité de nombreux ensembles de données publics volumineux, les progrès dans les architectures neuronales profondes, ainsi que le développement de nouveaux modèles génératifs. Il s'agit toujours d'un domaine de recherche actif, avec de nouveaux modèles et techniques en développement pour améliorer leurs performances et élargir leurs applications.

Les modèles génératifs peuvent être utilisés dans une grande variété d'applications, notamment la génération de données, la complétion de données (inpainting), la super-résolution, le transfert de style, la détection d'anomalies, l'adaptation de domaine, ainsi que l'apprentissage de représentations démêlées.

Les modèles génératifs ont le potentiel de révolutionner des industries telles que le divertissement, l'art, le design et la finance. Des produits récents d'IA tels que ChatGPT, Midjourney et Stable Diffusion ont démontré une efficacité et une diversité dans la génération de données. Cela marque une avancée significative dans le domaine de l'intelligence artificielle.

Dans cette thèse, notre principal objectif se concentre sur l'apprentissage de représentations démêlées et la génération conditionnelle.

Liste des contributions

Cette section vise à mettre en évidence les contributions apportées lors de ma thèse de doctorat. Les principales contributions de cette thèse sont les suivantes :

- Contributions bibliographiques
  - Nous présentons les modèles génératifs, y compris les auto-encodeurs variationnels (VAE), les réseaux antagonistes génératifs (GAN) et les modèles de diffusion.
  - Nous effectuons une revue approfondie de l'application de ces trois modèles génératifs dans l'apprentissage de représentations démêlées. De plus, nous regardons comment il est possible de générer conditionnellement des données avec ces modèles.
- Contributions méthodologiques
  - Nous proposons un auto-encodeur variationnel démêlé pour déterminer le sexe d'un individu à partir d'un maillage des os de sa hanche. Le modèle proposé permet, par construction, d'apporter une interprétation des résultats.
  - Nous introduisons deux nouvelles méthodes d'apprentissage de représentations démêlées qui encodent les facteurs de haut-niveau ainsi que leurs caractéristiques dans l'espace latent.
- Contributions applicatives
  - Nous démontrons le potentiel de l'apprentissage de représentations démêlées pour l'interprétation des images médicales.
  - La représentation démêlée proposée permet un contrôle précis des étiquettes et de leurs caractéristiques dans les images générées.
  - Nous vérifions l'adéquation des modèles de diffusion dans la génération de données séquentielles, telles que des séquences temporelles d'expressions faciales. De plus, en conditionnant le processus inverse du modèle de diffusion, il devient possible de gérer diverses tâches de génération conditionnelle.

Contenu du mémoire

Le mémoire est constitué de sept chapitres : une introduction, un état de l'art, quatre chapitres qui sont pour chacun d'entre eux en rapport avec un article, et une conclusion.

Le premier chapitre est un chapitre introductif qui présente rapidement le contexte de la thèse, à savoir les modèles génératifs. Les modèles génératifs sont des modèles d'apprentissage automatique qui permettent d'apprendre, dans un premier temps, la distribution d'un ensemble de données et, dans un second temps, de générer de nouveaux échantillons suivant la distribution apprise. Ces modèles suscitent un grand intérêt car ils ont obtenu des résultats très prometteurs dans de nombreuses disciplines : ils peuvent aujourd'hui produire des données hautement réalistes et diversifiées, ce qui est utile pour de nombreuses applications. On peut s'attendre à ce que ces modèles révolutionnent un large éventail de secteurs comme le divertissement, l'art, et la finance... Des produits récents, tels que ChatGPT, Midjourney et DALL-E illustrent très bien le potentiel de ces modèles. Les modèles génératifs représentent également un domaine de recherche très actif : de nouveaux modèles et de nouvelles techniques sont développés pour améliorer leurs performances et étendre leur champ d'application.

Le second chapitre présente un état de l'art des modèles génératifs, en mettant principalement l'accent sur les auto-encodeurs variationnels (VAEs), les réseaux antagonistes génératifs (GANs) et les modèles de diffusion, ainsi que leurs applications, notamment pour l'apprentissage de représentations démêlées et la génération conditionnelle.

Dans une première section, nous présentons une revue approfondie des modèles génératifs, en mettant particulièrement l'accent sur trois types de modèles génératifs : les VAEs, les GANs et les modèles de diffusion.

L'un des modèles génératifs les plus populaires est l'auto-encodeur variationnel (VAE) : il apprend une représentation de faible dimensionnalité des données d'entrée en les encodant dans un espace latent puis en les décodant de nouveau dans l'espace original. Le VAE se distingue des auto-encodeurs traditionnels par le fait qu'il apprend la distribution des données d'entrée. Pour cela, il se base sur les statistiques bayésiennes et en particulier les approches variationnelles. Cela lui permet notamment de générer de nouveaux échantillons à partir de la distribution apprise. Alors que les VAEs ont montré un grand succès dans la génération de nouveaux échantillons sur des petits ensembles de données simples, leurs performances sont limitées lorsqu'ils sont appliqués à des ensembles de données plus complexes, tels que des images naturelles. Dans de tels cas, les images générées sont souvent floues et manquent d'informations haute fréquence, ce qui constitue une critique courante des VAEs. Nous présentons alors différentes

stratégies qui ont été proposées de manière à améliorer ces performances. Nous finissons cette partie en décrivant les applications principales des VAEs.

Les modèles de diffusion ont connu une attention considérable ces dernières années. Comme les VAEs, Il s'agit également d'un modèle à espace latent mais les variables latentes correspondent à des versions bruitées des données d'origine : le processus de diffusion produit les versions bruitées (jusqu'à obtenir du bruit blanc) alors que le processus inverse les débruite. On peut noter qu'uniquement le processus inverse contient des paramètres à apprendre. Les modèles de diffusion sont particulièrement utiles pour générer des images de haute qualité et ont montré des performances supérieures à d'autres modèles génératifs pour une variété de tâches, telles que la génération d'images, la synthèse audio, la modélisation du texte et la génération de nuages de points. Malgré la qualité impressionnante des données générées, il y a tout de même deux limitations importantes : le temps d'échantillonnage est lent et il manque, en comparaison avec les VAEs, une fonctionnalité d'encodage des données (ce qui peut être problématique pour certaines applications). Nous décrivons rapidement dans cette partie comment le temps d'échantillonnage peut être accéléré.

Les GANs se composent de deux composants principaux : un réseau générateur et un réseau discriminateur. L'objectif du discriminateur est de distinguer correctement entre les échantillons réels et synthétiques, tandis que l'objectif du générateur est de produire des échantillons synthétiques indiscernables des échantillons réels. Les GANs peuvent générer efficacement des données synthétiques hautement réalistes, mais ils peuvent être difficiles à entraîner. Nous présentons différentes stratégies pour obtenir un apprentissage stable et efficace. Les progrès réalisés peuvent être attribués à deux facteurs clés : l'amélioration de la fonction de perte et de l'architecture des modèles.

Chaque modèle génératif a ses propres avantages et inconvénients. Par exemple, les VAEs bénéficient d'un encodage des données, mais ils ont tendance à perdre les informations haute fréquence des données. En revanche, les GANs ont la capacité de produire des images de haute qualité, mais ils sont difficiles à entraîner. Les modèles de diffusion, bien qu'ils soient capables de générer des images de haute qualité, impliquent un processus de génération complexe et souffrent d'un échantillonnage lent. Par conséquent, chaque méthode a ses propres scénarios d'application appropriés. Par exemple, les modèles de diffusion sont préférés lorsque la qualité de l'image prime sur le temps de génération. Les GAN sont bien adaptés aux applications en temps réel, tandis que les VAE se révèlent précieux pour traiter des données bruités. Dans le même temps, de nombreux chercheurs tentent de combiner ces modèles pour compenser leurs limitations individuelles. Nous présentons dans une dernière sous-section les différents modèles ainsi obtenus.

Après avoir introduit les trois principaux modèles génératifs, nous explorons, dans

une seconde section, l'une de leurs applications : l'apprentissage de représentations démêlées.

Démêler les facteurs de variation des données est un défi important dans les domaines de l'apprentissage automatique et de la vision par ordinateur. Une représentation démêlée consiste à encoder séparément des caractéristiques observables des données, de sorte que ces éléments possèdent une signification sémantique interprétable. Par exemple, lorsque qu'une caractéristique des données, telle que la couleur d'un visage, change, uniquement l'élément correspondant à cette caractéristique devrait changer. Ce processus vise à séparer les facteurs de variation des données, ce qui permet de les analyser et de les examiner de manière indépendante. Atteindre une représentation démêlée est un objectif important car cela peut améliorer l'interprétabilité, la contrôlabilité, la généralisabilité et la robustesse du modèle. Nous étudions dans cette section comment les VAEs, les GANs et les modèles de diffusion peuvent permettre d'obtenir des représentations démêlées (dans le cas supervisé ou non-supervisé).

La dernière section du chapitre 2 est finalement consacrée à la génération conditionnelle. Dans la section précédente, nous avons vu que les représentations démêlées peuvent être utilisées pour le contrôle des caractéristiques et la génération conditionnelle. Cependant, la génération conditionnelle peut également être réalisée sans faire appel à de telles représentations. De plus, à la place d'utiliser directement des étiquettes pour conditionner la génération, certaines approches utilisent des images ou du texte pour contrôler le processus de génération. Une approche consiste à utiliser d'autres images pour contrôler la génération d'images, ce qui permet des tâches telles que l'adaptation de domaines. D'autres approches impliquent l'utilisation de texte pour contrôler les caractéristiques d'une image. A l'inverse, il existe des méthodes qui utilisent des images pour contrôler la génération de texte. Dans cette section, nous explorons ces différents types de génération conditionnelle.

Les chapitres suivants sont dédiés à la présentation du travail de thèse.

Le troisième chapitre est constitué de l'article suivant :

Kaifeng Zou, Sylvain Faisan, Boyang Yu, Sébastien Valette, Hyewon Seo. "4D Facial Expression Diffusion Model". Soumis à ACM Transactions on Multimedia Computing, Communications, and Applications.

Dans cet article, nous avons proposé un modèle qui permet de générer des séquences d'expressions faciales tridimensionnelles conditionnellement à différents signaux. Le modèle génératif est basé sur un modèle de diffusion. De manière à capturer efficacement les caractéristiques temporelles des séquences, nous avons proposé d'utiliser un transformer bi-directionel pour former l'épine dorsale (backbone) du modèle de dif-

fusion. Bien que le modèle soit entraîné de manière inconditionnelle, son processus inverse peut être conditionné dans un second temps. Cela nous permet de développer différentes tâches incluant diverses générations conditionnelles (conditionnement par une étiquette, par du texte, par des séquences partielles ou simplement par une géométrie). Les expériences ont été menées sur deux ensembles de données : CoMA et BU4DFE. Les performances de l'approche ont été évaluées de la manière suivante : nous avons comparé notre méthode à ACTOR, Action2Motion et Motion3DGAN pour la génération conditionnellement à une étiquette. Pour la génération conditionnellement à un texte, nous avons comparé notre méthode à MotionCLIP. Le modèle proposé montre des très bons résultats et peut produire des maillages faciaux plausibles de divers types d'expressions sur différents sujets.

Le chapitre 4 est composé de l'article suivant :

Kaifeng Zou, Sylvain Faisan, Fabrice Heitz, Marie Epain, Pierre Croisille, Laurent Fanton, and Sébastien Valette. Disentangled representations: towards the interpretation of sex determination from the hip bone. The Visual Computer journal 2023.

Les méthodes de classification basées sur les réseaux de neurones sont souvent critiquées pour leur manque d'interprétabilité et d'explicabilité. En mettant en évidence les régions de l'image d'entrée qui contribuent le plus à la décision, les cartes de saillance sont devenues une méthode populaire pour rendre les réseaux de neurones interprétables. En imagerie médicale, elles ne semblent pas trop adaptées aux problèmes de classification pour lesquels les caractéristiques qui permettent de distinguer les classes sont spatialement corrélées. A noter que nos expériences ont été réalisées dans le cadre de la détermination automatique du sexe à partir des os de la hanche.

Nous proposons dans cet article un nouveau paradigme. Au lieu de chercher à comprendre ce que le réseau de neurones a appris ou comment la prédiction est réalisée, nous cherchons à révéler les différences entre les classes. Pour cela, l'échantillon analysé est transformé en le même échantillon mais appartenant à une autre classe. Ceci ouvre ainsi la voie à une interprétation plus facile des différences entre les classes. Par exemple, si le maillage d'entrée est celui d'un homme, sa reconstruction en tant qu'homme devrait être similaire au maillage original. En revanche, la reconstruction en tant que femme devrait présenter des différences interprétables dans des régions spécifiques. De plus, en comparant les deux reconstructions avec le maillage original pour plusieurs sujets, l'utilisateur peut obtenir un aperçu des différences morphologiques entre les os du bassin masculin et féminin.

Dans cette optique, nous avons proposé un auto-encodeur variationel démêlé (DVAE), qui permet de modéliser les maillages du bassin, et qui démêle le facteur d'intérêt (le sexe) des autres variables latentes. Cette représentation fournit non seulement la classe

d'un nouvel échantillon, mais peut également générer une reconstruction pour chaque classe. Les résultats obtenus sont cohérents avec les connaissances des experts. De plus, l'approche proposée permet de confirmer ou de doûter du choix du classifieur, ou éventuellement de le remettre en question. Enfin, notre étude démontre que l'utilisation de ces deux reconstructions pour entraîner un classificateur binaire permet d'améliorer le taux de bonne classification.

Le chapitre 4 est constitué de l'article :

Kaifeng Zou, Sylvain Faisan, Fabrice Heitz, Sébastien Valette. "Joint disentanglement of labels and their features with VAE." 2022 IEEE International Conference on Image Processing (ICIP). IEEE, 2022.

Le chapitre 5 est constitué de l'article :

Kaifeng Zou, Sylvain Faisan, Fabrice Heitz, Sébastien Valette. "Disentangling high-level factors and their features with Conditional Vector Quantized VAEs." Pattern Recognition Letters, 2023.

Le chapitre 5 contient également, dans sa dernière section, une analyse de sensibilité et une comparaison avec une méthode basée-GAN. A noter que cette section n'a pas été publiée.

Les chapitres 5 et 6 traitent de la même problématique, à savoir, le démêlage des étiquettes et de leurs caractéristiques par une approche basée VAE. La plupart des approches semi-supervisées qui cherchent à obtenir des représentations démêlées à l'aide d'auto-encodeurs variationnels divisent la représentation latente en deux composantes : la partie non interprétable et la partie démêlée qui modélise explicitement les facteurs d'intérêt. Chaque facteur d'intérêt est donc associé à une variable latente du même type. Par exemple, si l'étiquette d'intérêt se réfère aux lunettes (1 lorsque le sujet porte des lunettes, 0 sinon), il y aura une variable catégorielle dans l'espace latent qui code la présence ou l'absence de lunettes. Cependant, cette variable ne permet pas de modéliser les caractéristiques des lunettes (par exemple, forme/taille/couleur des lunettes), qui peuvent être soit perdues, soit au mieux entremêlées dans les autres variables latentes. Pour résoudre ce problème, il est nécessaire de modéliser conjointement les facteurs de haut niveau et leurs caractéristiques. Nous avons proposé deux approches basées sur des auto-encodeurs variationnels qui modélisent explicitement à la fois les facteurs de haut niveau et leurs caractéristiques associées.

Dans la première approche appelée JDVAE (Joint disentanglement of labels and their features with VAE, chapitre 5), nous avons proposé une nouvelle structure de dépendance conditionnelle où les étiquettes et leurs caractéristiques appartiennent à l'espace latent. Dans ce modèle, les lois a priori conditionnelles des caractéristiques

(étant données les étiquettes) doivent être correctement choisies pour assurer les propriétés de démêlement souhaitées. De plus, la fonction de perte est composée de deux divergences de Kullback-Leibler, qui doivent être pondérées différemment, afin d'obtenir des résultats satisfaisants. Cela rend l'approche difficile à utiliser.

Pour surmonter les limitations de l'approche précédente, nous avons proposé un nouveau modèle appelé CVQVAE (Conditional Vector Quantized VAE, chapitre 6). Les caractéristiques associées aux facteurs de haut niveau ne sont plus considérées comme des variables aléatoires sur lesquelles il est nécessaire d'intégrer. Au lieu de cela, chaque caractéristique est calculée (de manière déterministe) à partir des données d'entrée à l'aide d'un réseau de neurones dont les paramètres peuvent être estimés conjointement avec ceux du décodeur et de l'encodeur. Ces caractéristiques (ainsi que les étiquettes et les variables latentes) sont ensuite utilisées par le décodeur pour reconstruire les données. Cette approche s'inspire du VAE conditionnel (CVAE), à la différence que la variable conditionnelle est connue pour le CVAE et qu'elle est calculée pour le CVQVAE. Nous obtenons ainsi un modèle simplifié (sans loi a priori conditionnelle pour les caractéristiques, et une seule divergence de Kullback-Leibler dans la fonction de perte). De plus, pour améliorer la qualité des images générées et en particulier pour générer des images moins floues, la loi a priori gaussienne sur la représentation latente a été remplacée par une distribution catégorielle. Le modèle résultant est plus difficile à optimiser, mais nous contournons ce problème avec une procédure d'apprentissage en deux étapes.

Les deux méthodes sont validées sur le jeu de données CelebA et comparées avec des méthodes basées VAE. Les résultats obtenus avec JDVAE montrent l'intérêt de modéliser explicitement à la fois les étiquettes et leurs caractéristiques. De plus, ils montrent également l'avantage d'utiliser AdaIN et des tokens apprenables pour construire le décodeur : le premier permet d'améliorer la qualité des images générées tandis que le second favorise les propriétés de démêlement du modèle. Enfin, l'approche CVQVAE surpasse toutes les approches testées, tant en termes de démêlement que de qualité des images générées. De plus, nous montrons l'efficacité de notre modèle sur le jeu de données CheXpert : la pathologie peut être visualisée en comparant la reconstruction avec et sans la pathologie.

Finalement, dans le dernier chapitre, nous fournissons un bref résumé des principales contributions de la thèse, en mettant l'accent sur leur importance à la fois du point de vue applicatif et méthodologique. De plus, nous explorons les orientations potentielles pour des recherches futures dans le domaine des modèles génératifs, en nous concentrant spécifiquement sur les représentations démêlées et la génération conditionnelle.

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to several individuals and organizations who have played pivotal roles in my academic journey and made my PhD career truly meaningful.

First and foremost, I extend my sincere thanks to Professor Sylvain Fasain, Fabrice Heitz, and Sebastien Valette for their unwavering guidance, support, and mentorship throughout my academic endeavors. Their expertise and encouragement have been invaluable, shaping my growth as a researcher.

I am also deeply grateful to my friends who have been with me through thick and thin during this challenging journey. Your camaraderie and companionship have made the often arduous PhD path far more enjoyable and memorable.

To my parents, your unending love, belief in my potential, and unwavering support have been the bedrock upon which my academic pursuits have thrived. I am eternally grateful for everything you've done for me.

I must also extend my appreciation to HUST (Huazhong University of Science and Technology) in China for affording me the incredible opportunity to pursue my studies in France. This cross-cultural experience has enriched my academic and personal life in ways I could have never imagined.

Collectively, these individuals and institutions have been instrumental in shaping my academic journey, and for that, I am profoundly thankful.

TABLE OF CONTENTS

CHAPTER 1

Introduction

1.1 Generative models

1.1.1 What are generative models?

Artificial intelligence has become a recurring theme in contemporary science fiction movies, as seen in characters like Joi in "Blade Runner 2049" and Jarvis in "Iron Man." These portrayals often depict AI systems with abilities comparable to those of humans.

In recent years, with advancements in deep learning, researchers have been increasingly focused on creating digital humans using neural networks. Microsoft's Xiaoice is a notable example of this. Xiaoice utilizes automatic story analysis to select appropriate tones and characters, effectively completing the entire process of audio creation.

Deep learning has enabled AI systems to simulate human-like qualities and perform tasks that were once exclusive to humans. Key to this technology is generative AI, which involves analyzing vast amounts of data, learning patterns, and generating text, speech, and other media that even surpass human capabilities.

Generative models have emerged as prominent techniques in deep unsupervised learning over the past decade. This is largely due to their ability to effectively analyze and comprehend unlabeled data. These models capture the underlying data distribution $x \sim p_D(x)$ and use that knowledge to generate similar data points $\hat{x} \sim p_\theta(x)$, where θ represents the learnable parameters.

In contrast to early approaches that relied on energy functions for generating high-dimensional data, which often faced challenges in terms of generation efficiency and quality, recent years have witnessed significant advancements in generative models. These advancements have been driven by several factors, including the availability of numerous large public datasets and the advancements in deep neural architectures and different generative models.

Generative models, such as Variational Autoencoder (VAE) [124, 122], Generative Adversarial Network (GAN) [69], Energy-based Model (EBM) [130, 120], normalizing flow [49, 201], diffusion models [89, 231], have shown significant promise in various fields, including image generation, speech synthesis, and natural language processing,

and continue to be an active area of research, with new models and techniques being developed to improve their performance and broaden their applications.

These generative models have the potential to revolutionize industries such as entertainment, art, design, and finance. Recent AI products like ChatGPT, Midjourney, and Stable Diffusion have demonstrated the efficiency and diversity of data generation that surpasses human capabilities. This marks a significant advancement in the field of artificial intelligence.

As the field of deep learning continues to evolve and mature, generative models are poised to play an increasingly important role in shaping the future of AI. With ongoing research and development, we can expect to see new and innovative generative models emerge, enabling even more sophisticated and advanced applications of this powerful technology.

1.1.2 What can generative models do?

Generative models can be used in a wide variety of applications, including but not limited to data generation, data completion and inpainting, super-resolution, style transfer, anomaly detection, disentangled representation learning, domain adaptation.

Data Generation One of the primary functions of generative models is to generate new data samples that resemble a given dataset. These models learn the underlying distribution of the training data and can generate realistic samples that resemble to the training data. This capability has wide-ranging applications, including generating synthetic images, videos, and audio for artistic purposes, data augmentation in machine learning, and simulating data for training and testing purposes.

Data Completion and Inpainting Generative models can also be used for data completion and inpainting tasks. Given an incomplete or partially missing input, these models can generate plausible and coherent predictions to fill in the missing information. This has applications in image and video inpainting, where damaged or missing regions can be reconstructed using the learned generative model. In Chapter 3, we additionally explore a related application involving facial animation completion. Similarly, in the case of the hip bone, we employ a VAE to handle the completion and classification of missing data, which is presented in Chapter 4.

Super-resolution Generative models can enhance the resolution and quality of low-resolution images. By learning the underlying patterns and structures of high-resolution images, these models can generate sharper and more detailed versions of low-resolution inputs. Super-resolution techniques find applications in image and video enhancement, and surveillance systems.

Domain Adaptation Generative models can adapt models trained on one domain

to perform well on a different but related domain. By learning the underlying shared distribution between domains, these models can generate synthetic samples that bridge the gap between the source and target domains. Domain adaptation techniques are valuable in scenarios where labeled data in the target domain is limited or unavailable.

Anomaly Detection Generative models can be employed for anomaly detection tasks. By learning the distribution of a given dataset, these models can identify data points that are significantly far from the learned distribution. This can be applied in various domains, including fraud detection, cybersecurity, and medical diagnostics.

Disentangled Representation Learning Generative models can learn disentangled representations, where underlying factors of variation are separated and controlled independently. This allows for manipulating specific attributes or characteristics of generated samples while keeping other factors constant. Disentangled representations find applications in image editing, attribute transfer, and data analysis, enabling more fine-grained control over generated outputs.

In chapter 4, we propose the use of disentangled representations as a means to provide a comprehensive interpretation of sex determination from hip bone. We also extend the existing disentangled representation learning method and propose two novel methods, as discussed in Chapter 5 and Chapter 6.

Conditional Generation Generative models can be conditioned on additional information or constraints to generate samples that meet specific criteria. For example, images can be generated by leveraging text descriptions or class labels as conditioning factor. This enables controlled and targeted generation in various domains, including image synthesis, text-to-image, and image-to-image translation. In Chapter 3, we introduce a method for the conditional generation of facial expression.

In this thesis, our primary focus lies on disentangled representation learning and conditional generation. We explore these specific fields, exploring their concepts, methodologies, and applications. The Chapter 2 will dig deeper into these topics, offering more comprehensive insights and detailed analyses associated to disentangled representation learning and conditional generation.

1.2 List of contributions

This section aims to highlight the contributions made during my PhD, emphasizing the originality and significance of the work. The following are the key contributions of this thesis:

- Bibliographical

  - We provide a detailed explanation of generative models, including Variational Autoencoders (VAE), Generative Adversarial Networks (GAN), and diffusion models.
  - We conduct an in-depth review of the application of these three generative models in disentangled representation learning and conditional generation.

- Methodological
  - We propose a disentangled variational autoencoder for data-driven sex determination and interpretation.
  - We introduce two novel methods for disentangled representation learning that encode high-level factors and their features into the latent representation.

- Applicative
  - We demonstrate the potential of disentangled representation learning for the interpretation of medical images.
  - The proposed disetangled representation allows accurate control of labels and their features in generated images.
  - We verify the suitability of diffusion models in generating sequential data, such as facial expressions. Moreover, by conditioning the reverse process of the diffusion model, it becomes capable of handling diverse conditional generation tasks.

1.3 Thesis Structure

In order to provide a clear framework for the research study, the thesis is organized into the following chapters:

- **Chapter 2** This chapter presents a detailed explanation of VAE, GAN, and diffusion model, along with an in-depth review of the state-of-the-art in disentangled representation learning and conditional generation using these generative models.
- **Chapter 3** The focus of this chapter is on the use of the diffusion model for facial animation generation. The feasibility of employing the diffusion model for generating sequential data is validated, and a versatile framework is used: we train an unconditional model and subsequently condition the reverse process with various conditions to enable generation. This approach allows us to create versatile models, as we only need to train the diffusion model once and can then

condition it in a plug-and-play manner. The chapter 3 is mainly composed of this article:

Kaifeng Zou, Sylvain Faisan, Boyang Yu, Sébastien Valette, Hyewon Seo. 4D Facial Expression Diffusion Model

- **Chapter 4** This chapter introduces a supervised disentangled representation learning method for sex determination and interpretation. Initially, a disentangled Variational Autoencoder (VAE) is trained to generate hip bones for both sexes. The latent space of the VAE disentangles the identity information (z) and the sex information (y). By providing the hip bone of an individual as input, the disentangled VAE can be used to generate the hip bone of the same individual but for both sexes. A comparison is then conducted between the two generated hip bones to elucidate the distinctions in sex determination, specifically for individuals lacking medical expertise or knowledge.

  The Chapter 4 is mainly composed of the following article:

  Kaifeng Zou, Sylvain Faisan, Fabrice Heitz, Marie Epain, Pierre Croisille, Laurent Fanton, and Sébastien Valette. Disentangled representations: towards the interpretation of sex determination from the hip bone. The Visual Computer journal 2023.

- **Chapter 5, 6** When aiming for a supervised disentangled representation, a single label (high-level factor) can encompass a wide range of attributes and characteristics. For instance, when generating a human face with glasses, a smile, or a beard (labels), there are numerous possibilities for the specific types of glasses, smiles, and beards that can be incorporated. Extracting these specific features and representing them in disentangled forms becomes crucial.

  Chapter 5 and 6 in the thesis both aim to achieve a shared objective, which is to model the characteristics associated with specific labels.

  In Chapter 5, an innovative extension of the existing work of VAE [122] is presented. This approach provides a novel method to model the features associated to the high-level factor by introducing a variable into the latent space. On the other hand, Chapter 6 propose a method that enhances both the generation quality and accuracy.

  Notably, Chapter 5 is mainly composed of the article:

  K. Zou, S. Faisan, F. Heitz, , and S. Valette. Joint disentanglement of labels and their features with VAE. In IEEE International Conference on Image Processing (ICIP), 2022.

  Chapter 6 is mainly composed of the article:

K. Zou, S. Faisan, F. Heitz, and S. Valette. Disentangling high-level factors and their features with conditional vector quantized vaes. Pattern Recognition Letters, 2023.

Note that Chapter 6 also contains the supplementary material that has not been published.

- **Chapter 7** The final chapter concludes this thesis, followed by a discussion of the findings and potential future research directions.

CHAPTER 2

From Constraints to Creation: Disentangled Representations and Conditional Generation in State-of-the-Art Generative Models

2.1 Background

2.1.1 Variational Autoencoder

One of the most popular generative models is VAE [124], which learns a low-dimensional representation of the input data by encoding it into a latent space and then decoding it back to the original space. The VAE is different from traditional autoencoders in that it incorporates a probabilistic interpretation of the latent space, which allows for the generation of new data points by sampling from the learned distribution.

The fundamental concept of VAEs is to learn a probabilistic mapping between the observed data space x, and a latent space represented by z. The distribution of the latent space is associated with the corresponding data sample x. In this framework, the generative model learns a joint distribution, which can be expressed as follows:

$$p_\theta(x, z) = p_\theta(x \mid z)p(z), \tag{2.1}$$

where θ stands for learnable parameters. The latent variable z serves as the latent representation of the real data x and is endowed with a probabilistic interpretation, often assumed to follow a normal distribution ($p(z) \sim \mathcal{N}(0, I)$). $p_\theta(x|z)$ is often modeled as a Gaussian distribution, whose mean is given by a neural network with parameters θ. We have:

$$\begin{aligned} p_\theta(x|z) &= \mathcal{N}(x; f_\theta(z), vI) \\ &= \mathcal{N}(x; \hat{x}, vI), \end{aligned} \tag{2.2}$$

where the parameter v represents the variance, which is typically set as a hyperparameter with a value greater than zero. On the other hand, $\hat{x}$ is the mean of the distribution $p_\theta(x|z)$ which also refers to the reconstructed value of x.

However, computing the posterior distribution $p_\theta(z|x)$ directly is often computationally intractable. VAEs address this issue by employing the concept of variational

inference, which involves training a parametric inference model $q_\phi(z|x)$ to approximate the true posterior $p_\theta(z|x)$. $q_\phi(z|x)$ is defined as a Gaussian distribution whose mean and variance is estimated by a neural network with parameter ϕ, it writes:

$$q_\phi(z|x) \sim \mathcal{N}(z; \mu_\phi(x), \Sigma_\phi(x)). \tag{2.3}$$

Note that the distribution of $q_\phi(z|x)$ can be calculated with the same inference model for all values of x. The approach of sharing the variational parameters across all data points is known as amortized variational inference [67].

For any inference model, the likelihood $\log p_\theta(x)$ is written as:

$$\begin{aligned}
\log p_\theta(x) &= \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log p_\theta(x)\right] \\
&= \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left[\frac{p_\theta(x,z)}{p_\theta(z \mid x)}\right]\right] \\
&= \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left[\frac{p_\theta(x,z)}{q_\phi(z \mid x)}\frac{q_\phi(z \mid x)}{p_\theta(z \mid x)}\right]\right] \\
&= \underbrace{\mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left[\frac{p_\theta(x,z)}{q_\phi(z \mid x)}\right]\right]}_{=\mathcal{L}_{\theta,\phi}(x)(ELBO)} + \underbrace{\mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left[\frac{q_\phi(z \mid x)}{p_\theta(z \mid x)}\right]\right]}_{=D_{KL}\left(q_\phi(z|x)|p_\theta(z|x)\right)},
\end{aligned} \tag{2.4}$$

where the Kullback-Leibler (KL) divergence quantifies the difference between two distributions. Since the KL divergence is always non-negative, the first term of Eq. 2.4 is known as the Evidence Lower BOund (ELBO). As shown in Eq.2.4, the maximizing ELBO involves maximizing the marginal likelihood, $p_\theta(x)$, and minimizing the KL divergence between the estimated posterior, $q_\theta(z|x)$, and the true posterior $p_\theta(z|x)$. By writing $p_\theta(x,z) = p(z)p_\theta(x|z)$, the ELBO term becomes:

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)). \tag{2.5}$$

It is worth noting that the parameters θ and ϕ can be optimized together. The first term, $\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$, is estimated using a Monte Carlo method, specifically the Stochastic Gradient Variational Bayes (SGVB) algorithm, which incorporates the reparametrization trick. The second term can be computed analytically since both $q_\phi(z|x)$ and $p(z)$ follow the Gaussian distribution.

Another perspective to comprehend VAEs is by considering two joint distributions. The first joint distribution, denoted as $p_\theta(x,z)$, captures the relationship between the generated data x and the latent variable z, as shown in Equation (2.1). Similarly, the second joint distribution, denoted as $q_\phi(x,z)$, accounts for the approximation of the

latent variable and the observed data x, and can be expressed as:

$$q_\phi(x, z) = q_\phi(z \mid x) q_D(x), \tag{2.6}$$

where $q_D(x)$ represents the empirical (data) distribution which is a mixture distribution:

$$q_D(x) = \frac{1}{N} \sum_{i=1}^{N} q_D^{(i)}(x), \tag{2.7}$$

where N is the size of the dataset and each component $q_D^{(i)}(x)$ represents a distribution that can be described as a Dirac delta function centered at the value $x_{(i)}$ for continuous data ($x_{(i)}$ is the i-th sample of dataset), or a discrete distribution where all the probability is concentrated at the value $x_{(i)}$ for discrete data.

Maximizing ELBO is equivalent to minimizing the KL divergence between $q_\phi(x, z)$ and $p_\theta(x, z)$. This equivalence is demonstrated in Appendix A.

$$\begin{aligned} D_{KL}(q_\phi(x, z) \| p_\theta(x, z)) &= \mathbb{E}_{q_D(x)} \left[\mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x \mid z)] + D_{KL}(q_\phi(z \mid x) \| p(z)) \right] + C \\ &= -\mathcal{L}_{\theta,\phi}(x) + C. \end{aligned} \tag{2.8}$$

Fig. 2.1 Image generation results with VAE on CelebA dataset [150]

Limitation and improvements While VAEs have shown great success in generating new samples on small datasets, their performance tends to suffer when applied to more

complex datasets, such as natural images. In such cases, several examples showcasing the results of a standard VAE are presented in Figure 2.1. We can observe that the generated images are often blurry and lack some high-frequency information, which is a common criticism of VAEs.

There is an inherent trade-off between compression and reconstruction accuracy, as discussed in [7]. Remarkably, the two terms in the loss function presented in Eq. (2.5) exhibit a fundamental contradiction. Specifically, when the first term of Eq. (2.5) ($\mathbb{E}_{z\sim q(z|x)}[\log p_\theta(x \mid z)]$) is too large, it results in a latent space that lacks diversity. Conversely, if the second term ($D_{KL}(q_\phi(z \mid x)||p(z))$) is too small, the latent space becomes excessively random, resulting in inaccurate reconstructions. Therefore, finding a balance between these two losses is crucial for generating high-quality outputs in a VAE.

A common approach is to alter the variance of $p_\theta(x|z)$. In the standard VAE, $p_\theta(x|z)$ is typically assumed to follow a Gaussian distribution with a fixed variance denoted as v (as shown in Eq. 2.2), which becomes a hyperparameter. Various methods have been proposed to estimate the variance [208]. These include obtaining it from the output of a neural network, calculating it for each mini-batch, or treating it as a learnable parameter.

An equivalent approach is to reformulate the ELBO (Eq. 2.5) as a combination of the reconstruction loss and a weighted KL divergence loss (v is fixed). To control the significance of the KL divergence loss, a parameter β is typically introduced, as discussed in [86].

Another way to improve the performance of VAEs is to a use different regularisation in the latent space to improve the quality of generated samples, such as the wasserstein distance [239] or vector quantization [246, 199]. Note that Vector Quantized Variational Autoencoder (VQVAE) is a variant of the traditional VAE that replaces the continuous Gaussian distribution of the latent variables with a discrete distribution. The encoder maps the input data to a sequence of discrete latent codes, which are then quantized to a codebook of learned discrete embeddings. The decoder then maps the discrete codes back to the original input space, producing a reconstructed output. The quantization step enforces a form of discretization in the latent space, which encourages the model to capture the underlying structure and dependencies of the data.

It is worth noting that Hierarchical VAEs are another improvement of VAEs which use a hierarchical framework for capturing the underlying data distribution. This architecture consists of multiple levels, with each level of the encoder and decoder modeling a specific level of abstraction of the data. Ladder VAE [227] utilizes lateral connections between intermediate latent variables across layers to achieve a hierarchical architecture. NVAE [244] achieves very high-quality and high-resoluition image generation by a deep hierarchical VAE that uses depthwise separable convolutions for the generative

model and regular convolutions for the encoder model. The architecture and results are shown in Fig.2.2

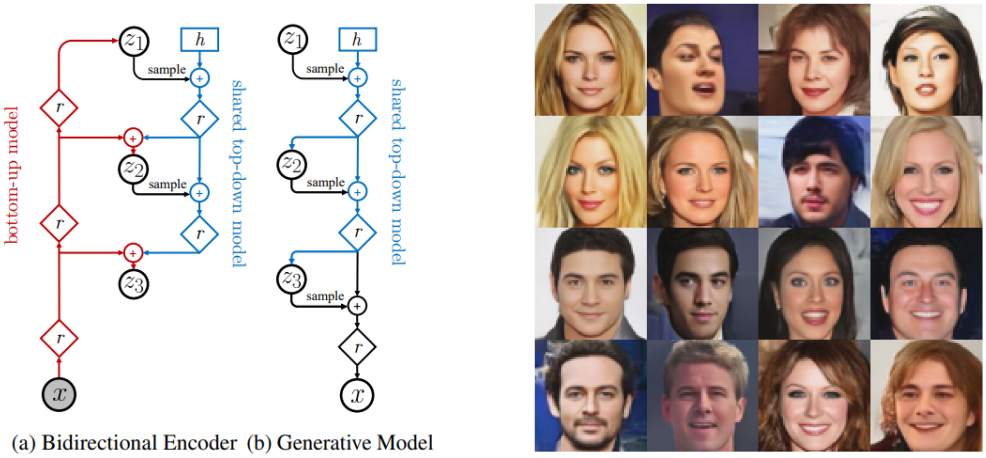


Fig. 2.2 On the left side, we can see the hierarchical architecture of NVAE, where h and r denote the trainable parameters and residual block, respectively. On the right side, we showcase the results of NVAE on the CelebA dataset. Drawn from [244]

Applications VAEs have been used in a wide range of applications in machine learning and computer vision. One of the most common applications of VAEs is in the field of data generation. By learning the underlying distribution of a dataset, VAEs can generate new data samples that are similar to the original data. This has been demonstrated in various domains such as generating realistic faces [270], text [23], speech [38], and human motion [182]. The ability to generate new data with similar characteristics as the original data has led to the development of creative applications such as image synthesis and art generation.

In addition to data generation, VAEs can also be used for image reconstruction tasks due to their encoding ability such as image denoising [102], image inpainting [180], and image super-resolution [151].

Another application of VAEs is anomaly detection, where the model is trained on normal data and then detects anomalies as inputs that do not fit the learned distribution. This has been applied in various fields such as fraud detection [238], medical diagnosis [79], and cybersecurity [234]. By detecting outliers in data, VAEs can help identify potentially problematic situations.

Finally, VAEs have found applications in various other scenarios, such as dimensionality reduction [71] and reinforcement learning [166]. The ability of VAEs to learn a compressed representation of high-dimensional data makes them useful for reducing the complexity of datasets. Additionally, their use in reinforcement learning tasks involves using the encoded latent space representation to make decisions about the environment.

Overall, VAEs have proven to be a powerful tool in a variety of applications, demonstrating their versatility and usefulness in the field of machine learning. With ongoing research and development, it is likely that the applications of VAEs will continue to expand, further establishing their importance in the field.

2.1.2 Diffusion Models

Regarding diffusion models, commonly cited techniques include energy-based models [87], score matching [101], and Langevin dynamics. In brief, this approach involves training energy-based models utilizing methods like score matching and subsequently using Langevin equations for sampling from these models [229, 65, 266, 224]. Theoretically, this method is a well-established solution that holds the potential for generating and sampling any continuous object, such as images and speech. However, practically speaking, energy function training proves to be a challenging task, especially with high-dimensional data like high-resolution images. Achieving complete energy function training is difficult. Moreover, there is a high level of uncertainty when using Langevin equations for sampling from the energy model, often leading to noisy results.

For a significant period, the conventional path of diffusion models involved experimenting solely with low-resolution images. However, the recent upsurge in diffusion models' popularity is primarily due to the Denoise Diffusion Probabilistic Model (DDPM) proposed in 2020 [89]. Notably, the mathematical framework behind DDPM was introduced earlier in 2015 [224]. Nevertheless, it was only with the DDPM that high-resolution image generation became possible. While DDPM also adopts the name *diffusion model*, it is fundamentally different from traditional models that rely on Langevin equation sampling, except for a few similarities in their sampling process. In my opinion, it is even more closely related to VAE. In any case, DDPM marks a new beginning and a new chapter in this field.

From VAE to DDPM In the traditional VAE (as discussed in Sec. 2.1.1), the encoding and generating processes are one-step processes, which can be represented as:

$$\textbf{Encoding} : z = f(x), \textbf{Generating} : x = g(z). \tag{2.9}$$

The VAE framework revolves around three distributions: the encoding distribution $q_\phi(z|x)$, the generating distribution $p_\theta(x|z)$, and the prior distribution $p(z)$. One of the key advantages of VAE is its ability to generate data while also having the capability to encode input data x into a latent representation z.

Despite its relatively straightforward structure and the existence of a mapping relationship between x and z, VAE's expressive power is limited due to its inherent challenge in accurately modeling probability distributions. One common criticism of VAE

is its tendency to produce blurry generated results, as illustrated in Figure 2.1.

DDPM is a similar approach that significantly enhances the quality of generation (see Fig.2.3). It achieves this by dividing the encoding and generating processes into T distinct steps. It can be modeled as follows.

$$\begin{aligned} \textbf{Encoding} &: z = x_0 \to x_1 \to x_2 \to x_3 \to \cdots \to x_{T-1} \to x_T, \\ \textbf{Generating} &: x = x_T \to x_{T-1} \to \cdots \to x_3 \to x_2 \to x_1 \to x_0, \end{aligned} \tag{2.10}$$

where x_0 represents the true data. The encoding process involves gradually introducing noise to the data. We apply a gradual perturbation to the original data x_0 until we obtain x_T from a Gaussian distribution $\mathcal{N}(0, I)$. The generation process involves removing noise from x_T and gradually recovering x_0 through a series of T iterations. Each encoding process is represented by $q(x_t|x_{t-1})$, while each generation process is represented by $p(x_{t-1}|x_t)$.

In this framework, each state transition, namely $p(x_t|x_{t-1})$ and $q(x_{t-1}|x_t)$, models a minor change in the process, which can be approximated by a Gaussian distribution. The joint distribution corresponding to these transitions is expressed as follows:

$$q\left(x_0, \cdots, x_T\right) = q\left(x_T \mid x_{T-1}\right) \cdots q\left(x_1 \mid x_0\right) \tilde{q}\left(x_0\right), \tag{2.11}$$

$$p\left(x_0, \cdots, x_T\right) = p\left(x_0 \mid x_1\right) \cdots p\left(x_{T-1} \mid x_T\right) p\left(x_T\right), \tag{2.12}$$

where $\tilde{q}\left(x_0\right)$ is the data distribution. In DDPM, both the diffusion process and the reverse process are represented as a Markov chain. Equation 2.11 describes the diffusion process which is determined by the predefined noise schedule parameters, namely α and β. The state transition of the diffusion process can be defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, \beta_t I). \tag{2.13}$$

This equation specifies that the distribution of x_t given x_{t-1} follows a Gaussian distribution. The mean of this distribution is obtained by scaling the previous input x_{t-1} with the scalar $\sqrt{\alpha_t}$, and the covariance is $\beta_t I$, where I is the identity matrix. As in VAE, we also use the reparametrization trick to represent each variable. Thus we have:

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{\beta_t}\varepsilon_t, \tag{2.14}$$

where $\varepsilon_t \sim \mathcal{N}(0, I)$

Another important feature is that we are able to directly calculate x_t from x_0 which

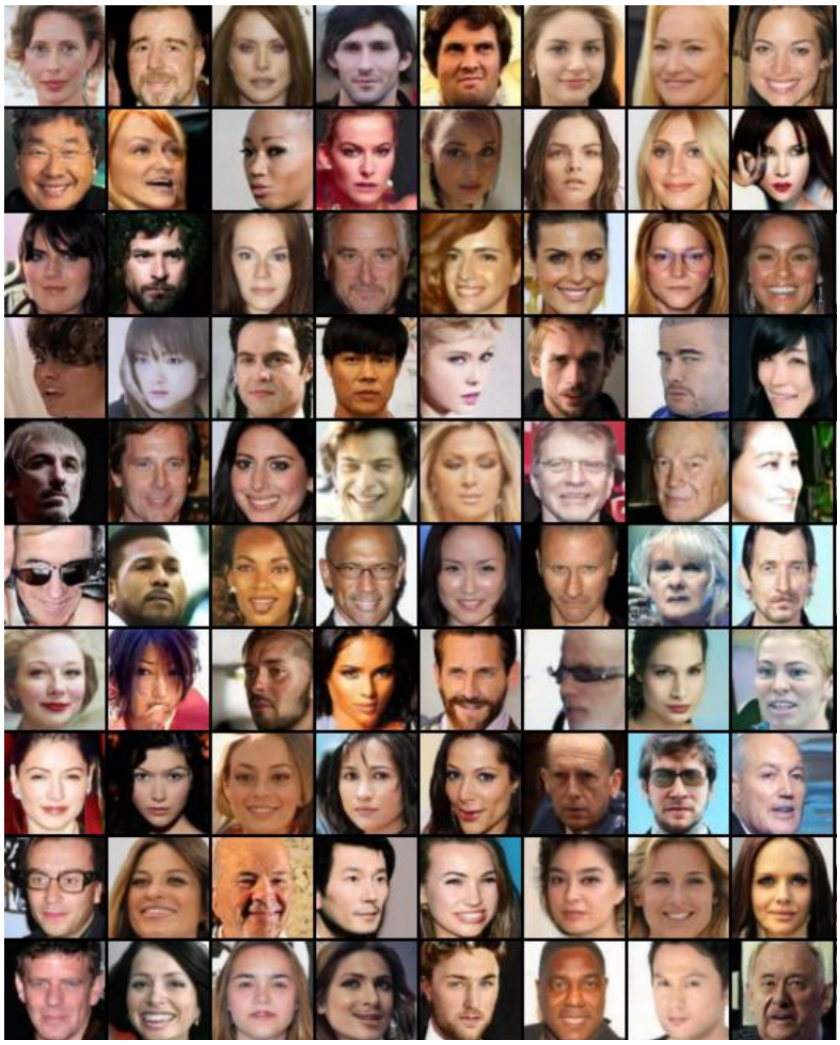


Fig. 2.3 Image generation results with DDPM on CelebA dataset. Image taken from [177].

allows us to train DDPM at any time step t. To simplify the calculation of x_t, we need to set $\alpha_t + \beta_t = 1$ for each time step t. This allows us to express $q(x_t \mid x_0)$ in a more convenient form (demonstration is shown in Appendix B):

$$\begin{aligned} q(x_t|x_0) &= \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, \bar{\beta}_t I) \\ x_t &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{\bar{\beta}_t}\varepsilon_t \end{aligned} \tag{2.15}$$

where $\varepsilon_t \sim \mathcal{N}(0, I)$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\bar{\beta}_t = 1 - \bar{\alpha}_t$. The second equation of Eq.2.15 is the reparameterization of x_t.

Therefore, in the context of DDPM, the mean and variance for each step x_t are predetermined. Unlike traditional VAEs that learn the mean and variance through neural networks, DDPM focuses solely on the generative process by discarding the encoding process. Note that only the reverse process contains trainable parameters θ so that p will

be denoted p_θ. For the generation process (reverse process), the mean of each denoising step $p_\theta(x_{t-1}|x_t)$ is learned by a neural network μ_θ, and is defined as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t), \sigma_t^2 I), \tag{2.16}$$

where θ represents learnable parameters and σ_t is predetermined. The setting of σ is discussed in Appendix B.

The whole process can be described by a directed graphical model shown in Fig.2.4

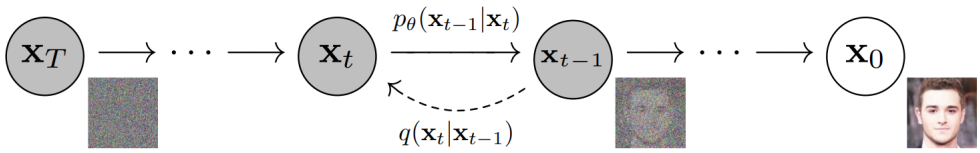


Fig. 2.4 The directed graphical model of DDPM. Image is taken from [89]

In order to train the diffusion model, the optimization objective for DDPM is to minimize the Kullback-Leibler (KL) divergence between the two joint distributions: p_θ and q. This is similar to the optimization objective for VAE. The optimization objective for DDPM is:

$$\begin{aligned} D_{KL}(q||p_\theta) = \int & q\left(x_T \mid x_{T-1}\right) \cdots q\left(x_1 \mid x_0\right) \tilde{q}\left(x_0\right) \\ \log & \frac{q\left(x_T \mid x_{T-1}\right) \cdots q\left(x_1 \mid x_0\right) \tilde{q}\left(x_0\right)}{p_\theta\left(x_0 \mid x_1\right) \cdots p_\theta\left(x_{T-1} \mid x_T\right) p_\theta\left(x_T\right)} dx_0 dx_1 \cdots dx_T. \end{aligned} \tag{2.17}$$

Since q does not contain any trainable parameters, the objective Eq. (2.17) can be written as follows:

$$\begin{aligned} & -\int q\left(x_T \mid x_{T-1}\right) \cdots \tilde{q}\left(x_0\right) \log(p_\theta\left(x_0 \mid x_1\right) \cdots p_\theta\left(x_T\right)) dx_0 \cdots dx_T \\ = & -\int q\left(x_T \mid x_{T-1}\right) \cdots \tilde{q}\left(x_0\right) \left[\log p\left(x_T\right) + \sum_{t=1}^{T} \log p_\theta\left(x_{t-1} \mid x_t\right)\right] dx_0 \cdots dx_T. \end{aligned} \tag{2.18}$$

Since $x_T \sim \mathcal{N}(0, I)$, the contribution of the term of $\log p(x_T)$ can be regarded as constant, and we can focus on the optimization of the remaining terms in the objective function. However, computing every step of the reverse process from scratch during training is computationally expensive. Therefore, the most efficient way to train a diffusion model is to optimize each term of the objective separately.

If we consider the utilization of a neural network ϵ_θ to estimate the noise associated with each step of the diffusion model, we can derive the straightforward objective

presented in DDPM [89] as follows:

$$\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I),x_0\sim\tilde{q}(x_0)}\left[\left\|\varepsilon-\epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0+\sqrt{\bar{\beta}_t}\varepsilon,t\right)\right\|^2\right] \tag{2.19}$$

Considering the specification of the noise approximator, we can derive the equation for x_{t-1} given x_t in the reverse process as follows:

$$\mu_\theta(x_t)=\frac{1}{\sqrt{\alpha_t}}(x_t-\frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t,t)). \tag{2.20}$$

Note that the integration of Equation 2.18 can be computed for each time step t as outlined in Appendix C.

Limitations and improvements Despite the impressive quality of images generated by the diffusion model (see Fig. 2.4), its slow sampling time and the lack of encoding functionality have limited its practical application. To address these limitations, researchers have proposed several approaches to enhance its performance. One such approach is DDIM [228], which accelerates the sampling process by reducing the number of required sampling steps. DDIM also allows for deterministic reverse processing, which means that we can determine the initial noise of the desired images, enabling various image editing possibilities, such as modifying image conditions [84]. However, DDIM inversion can result in instability and distorted reconstructions. To address this issue, [250] suggests a novel approach inspired by coupling layers in normalizing flow models [48], which provides mathematically exact inversion.

In addition, there are several tricks for training the diffusion model that can improve its performance, such as learning the variances of the reverse diffusion process, using a cosine noise schedule, adding extra loss terms to optimize the variational lower-bound. [170].

Applications Diffusion models have a wide range of applications across various fields, including computer vision, natural language processing, and audio signal processing, and continue to set new state-of-the-art (SOTA) records.

In the field of computer vision, diffusion models have demonstrated impressive capabilities for super-resolution [135, 212, 90], image inpainting [154], image translation [210, 188], and semantic segmentation [17, 24, 70]. But diffusion models are not limited to 2D image data alone. They are also capable of handling 3D data for point cloud generation and completion [293, 156, 157], as well as time series data for human motion generation [237, 286], face expression generation [303], time series forecasting and imputation [6], and video generation [81, 92, 273, 88].

In the field of natural language processing (NLP), diffusion models have also shown

great promise in the field of NLP, with numerous applications and use cases [13, 139, 31, 68, 47]. Additionally, diffusion models have proven to be particularly useful in the area of multi-modal learning, with one of the most popular applications being text-to-image synthesis [205, 14, 193, 169, 72, 245]. Among them, Stable Diffusion [205] has become the most widely used text-to-image model in both industry and people's daily life. Moreover, diffusion models have also been applied to other multi-modal tasks, such as text-to-video [222, 190], text-to-audio [186, 271, 233], and text-to-3D generation [267, 142, 185], all of which have achieved remarkable success. Furthermore, diffusion models have exhibited their versatility and broad applicability beyond the domains previously mentioned, including but not limited to molecular graph modeling [109, 94], medical image reconstruction [230], and robust learning [171]. The success and effectiveness of diffusion models across such a wide range of applications highlights their flexibility and wide-ranging utility in various problem domains.

2.1.3 Generative Adversarial Network

Generative Adversarial Networks, or GANs for short proposed by [69], are a type of neural network used for generative modeling. They consist of two main components: a generator network G and a discriminator network D. The generator network takes as input a random noise vector z and produces a synthetic sample $x' = G(z)$ that is intended to resemble real data samples x drawn from a training set.

The discriminator network, on the other hand, takes as input a sample x (either real or synthetic) and outputs a scalar value $D(x) \in [0, 1]$ representing the probability that x is a real sample. The goal of the discriminator is to correctly distinguish between real and synthetic samples, while the goal of the generator is to generate synthetic samples that are indistinguishable from real samples. The architecture of GAN is shown in Fig. 2.5.

The training process for GANs involves alternating between updating the generator and discriminator networks. Specifically, given a batch of training data $x_1, \ldots, x_m$ and a batch of noise vectors $z_1, \ldots, z_m$, the generator is updated by minimizing the following objective:

$$\min_G \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D\left(G\left(z_i\right)\right)\right). \tag{2.21}$$

Intuitively, this objective encourages the generator to produce synthetic samples that the discriminator is likely to mistake for real samples. Meanwhile, the discriminator is updated by maximizing the following objective:

$$\max_D \frac{1}{m} \sum_{i=1}^{m} \left[\log \left(D\left(x_i\right)\right) + \log \left(1 - D\left(G\left(z_i\right)\right)\right)\right]. \tag{2.22}$$

This objective encourages the discriminator to correctly distinguish between real and synthetic samples. Together, these two objectives create a "game" between the generator and discriminator, where the generator tries to produce samples that can fool the discriminator, and the discriminator tries to correctly classify samples as real or synthetic. Thus, the whole objective can be rewrite as follows.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \tag{2.23}$$

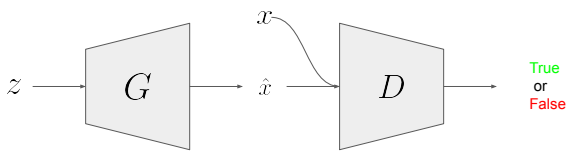


Fig. 2.5 The architecture of GAN. The generator G and the discriminator D are optimized alternatively.

Limitation and improvements GANs can efficiently generate highly realistic synthetic data, but they can be difficult to train due to the potential for the generator and discriminator to become stuck in a "stalemate". One major challenge is the optimization of the traditional GAN objective function, which involves minimizing the Jensen-Shannon divergence between the real and synthetic data distributions (see next paragraph), leading to instability. Additionally, the discriminator can become ineffective if the probability values it produces become too extreme, causing vanishing gradients and saturation.

Assuming the existence of an optimal discriminator $D^*(x)$, which can accurately distinguish between data samples x drawn from the true distribution $p_r(x)$ and those generated from the distribution $p_g(x)$, we can derive the optimal discriminator by setting the derivative of the discriminator loss function Eq. (2.22) to zero. It writes:

$$D^*(x) = \frac{p_r(x)}{p_r(x) + p_g(x)}. \tag{2.24}$$

By substituting this optimal discriminator into the generator loss function, referenced as Eq. (2.21), we can get:

$$\mathbb{E}_{x\sim p_r} \log \frac{p_r(x)}{\frac{1}{2}\left[p_r(x) + p_g(x)\right]} + \mathbb{E}_{x\sim p_g} \log \frac{p_g(x)}{\frac{1}{2}\left[p_r(x) + p_g(x)\right]} - 2\log 2. \tag{2.25}$$

It is equivalent to minimizing the Jensen-Shannon divergence:

$$D_{JS}\left(p_r, p_g\right) = D_{KL}\left(p_r \| p_m\right) + D_{KL}\left(p_g \| p_m\right), \tag{2.26}$$

where $p_m = (p_r + p_g)/2$. This aligns with the objective of generating synthetic data that is as close as possible to the real data distribution. However, the Jensen-Shannon estimate tends to remain constant or increase instead of decreasing. This occurs when the discriminator performs too well, and the gap between the p_r and p_g distributions is too large, resulting in the Jensen-Shannon distance approaching its maximum value of $\log 2$. As a consequence, the Jensen-Shannon distance saturates, the discriminator loss becomes zero, and the generated samples become meaningful in some instances, while in others, they collapse into meaningless images.

Due to the reasons mentioned above, achieving stable and effective training of GANs has been a hot topic in the research community. DCGAN [192] proposed architectural improvements to enhance the stability of GAN training. These improvements include using Average pooling and stride operations for downsampling in the network and ConvTranspose2D layers with stride for upsampling. The generator and discriminator are designed with specific architectures to optimize their performance. The generator utilizes the Tanh activation function for its output, while the discriminator employs LeakyReLU activation functions for each layer. Furthermore, fully connected hidden layers are removed to facilitate deeper architectures.

[213] introduced the concept of feature matching as a training approach for the generator. The idea is to minimize the distance between the features extracted by the discriminator from both the generated images and the real images. By aligning the features, the generator can learn to generate samples that resemble the real data distribution. Additionally, the paper proposes a technique called one-sided label smoothing as an alternative to the traditional binary labels used in GANs. Instead of assigning binary values (0 or 1) to real and fake samples, one-sided label smoothing assigns a lower value (e.g., 0.9) to the real samples. This modification helps prevent the discriminator from becoming overly confident, leading to more stable GAN training. Furthermore, the paper discusses several other techniques to address different challenges in GAN training. These techniques include virtual batch normalization, historical averaging, and minibatch discrimination. Virtual batch normalization aims to reduce internal covariate shift by normalizing the generator's intermediate layers using statistics from a reference batch. Historical averaging involves maintaining a running average of the generator's parameters to stabilize training and improve sample quality. Minibatch discrimination is a method to encourage diversity in generated samples by introducing additional information about the entire minibatch during the discriminator's computation. These

techniques collectively contribute to improving GAN training by addressing issues related to internal covariate shift, discriminator robustness, sample diversity, and overall stability.

In addition to the aforementioned techniques, a variant of GANs called the Wasserstein Generative Adversarial Network (WGAN) was introduced by [9]. Instead of using the Jensen-Shannon divergence, WGANs use the Wasserstein distance (also known as the earth mover's distance) to measure the distance between the real and synthetic data distributions.

$$W\left(p_r, p_g\right) = \inf_{\gamma \in \Pi\left(p_r, p_g\right)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|], \tag{2.27}$$

where $\Pi\left(p_r, p_g\right)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively p_r and p_g. The Wasserstein distance has several advantages over the Jensen-Shannon divergence, including being smoother and more stable to optimize. Additionally, WGANs use a modified discriminator that produces a scalar output rather than a probability value, making it less prone to vanishing gradients. Additionally, the concept of WGAN was expanded upon by WGAN Gradient Penalty (WGAN-GP) [75], which introduced a gradient penalty (GP) term in the discriminator to enforce the 1-Lipschitz constraint.

Moreover, Boundary Equilibrium GAN (BEGAN) [19] uses an equilibrium between the generator and discriminator to control the trade-off between image quality and diversity. The model aims to find a "boundary" where the generator produces images that are both high-quality and diverse. To achieve this, BEGAN introduces a new loss function based on the Wasserstein distance between the real and generated images, and adds a new parameter called the "equilibrium factor" that controls the balance between the generator and discriminator. The equilibrium factor is updated during training to ensure that the generator produces images that are diverse and of high quality.

In addition to improving its objective, GAN has also undergone numerous structural enhancements. Progressively-Growing GAN (PGGAN) (proposed by [113]) adopts a multi-scale GAN architecture in which both the generator (G) and discriminator (D) begin training with low-resolution images (e.g. 4x4) and gradually increase in depth by adding new layers during the training process. This leads to the generation of high-resolution images (e.g. 1024x1024) with sharp details.

StyleGAN [115] expands upon PGGAN and introduces novel architecture which leads to an automatically learned, unsupervised separation of high-level attributes and stochastic variation in the generated images. One of the key innovations of StyleGAN is the incorporation of a mapping network that transforms the input latent code into an intermediate latent code. This intermediate latent code allows for more fine-grained control over the generated images by disentangling different aspects of the image synthesis

process. Furthermore, StyleGAN incorporates an affine transformation that operates on the intermediate latent code. This transformation produces styles that influence the layers of the synthesis network using a technique called Adaptive Instance Normalization (AdaIN) [99]. AdaIN scales the normalized input with style spatial statistics, providing fine-grained control over specific image features. The unique architecture of StyleGAN has made it a widely recognized and influential approach in the field of generative adversarial networks. The architecture is shown in Fig.2.6.

The progress in GAN models can be attributed to two key factors: the improvement of GAN losses such as WGAN and WGAN-GP, and the enhancement of model architectures like PGGAN and StyleGAN. These advancements have allowed researchers and practitioners to explore the frontiers of generative modeling by generating data samples that are more realistic and diverse than ever before. As a result, GANs have gained immense popularity and have found extensive applications due to their remarkable adaptability to various neural network structures. The following section will highlight some of the applications of GANs.

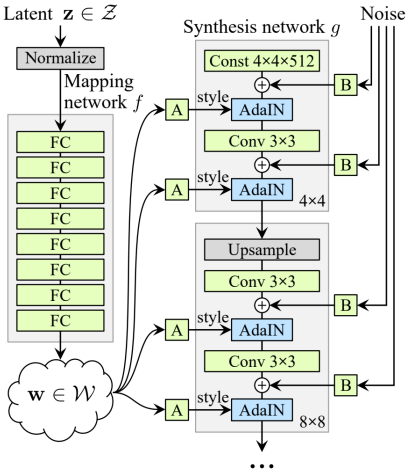


Fig. 2.6 Architecture of StyleGAN. Grawn from [115]

Applications GANs are incredibly powerful generative models capable of producing realistic samples that closely resemble the data they were trained on. This unique capability has led to their adoption across a wide range of fields within computer vision (CV) and artificial intelligence (AI). In particular, GANs have found numerous applications in various domains, including image, audio, and video. Thanks to their extensive development history and widespread adoption, GANs have become a go-to tool for many different applications. In table 2.1., we highlight some of the most popular GAN applications across different domains.

2.1.4 Bridging the Gap: Exploring the Relationship between VAE, GAN, and DDPM

Each model has its own advantages and disadvantages. VAE has nice encoding capabilities; however, it tends to lose high-frequency information of images. On the other hand, GANs have the ability to produce high-quality images, but they are challenging to train and often prone to mode collapse, which is when the generator only produces a limited number of outputs instead of the variety that is desired. Diffusion models,

Table 2.1 Summary of some the applications utilizing GANs.

Primary	Secondary	Papers
Computer Vision	Image Generation	[192],[16] [113], [115], [25], [263], [116], [114], [282]
	Image translation	[105], [296], [277], [121], [252], [36], [100], [37]
	Image in-painting	[50], [145]
	Facial landmark detection	[52], [275]
	Image super-resolution	[27], [299], [131], [257], [73]
	Facial attribute manipula-tion	[265], [264], [204], [276],[96]
	Text to image	[200], [283], [268],[284], [298], [290], [235], [64]
	Medical image	[80], [163], [78], [243], [214] , [269]
Video	2D video	[249], [242],[43], [39], [174]
	3D video	[175]
Audio	Language and Speech synthesis	[143], [95] [218]
	Music generation	[162], [281], [74], [97], [137]

while capable of generating high-quality images, involve a complex generation process and suffer from slow sampling. Therefore, each method has its own suitable application scenarios. For instance, diffusion models are preferred when prioritizing image quality over generation time. GANs are well-suited for real-time applications, whereas VAEs prove valuable in handling high-frequency noise within images. Simultaneously, many researchers are attempting to integrate these models to compensate for their individual limitations.

One promising approach that has gained significant attention is the combination of VAEs and GANs, known as VAE-GAN [129]. VAEs can struggle to capture the intricate details and textures of the input data, leading to blurry or low-resolution outputs. An improved method for image generation that addresses this issue is the use of adversarial training [160, 98, 129] which incorporates an adversarial loss term that encourages the generated images to match the distribution of the input data more closely. This hybrid model combines the encoding capabilities of VAEs and the image generation prowess

of GANs to overcome their individual limitations.

In VAE-GAN, the VAE component plays a crucial role in encoding input data into a latent space representation. This encoding process enables efficient compression and noise reduction, effectively capturing the essential features of the input. On the other hand, the GAN component focuses on generating high-quality images from the learned latent space representation. By leveraging the adversarial training framework, the GAN component learns to generate images that resemble the training data distribution, resulting in a realistic generation. The architecture of VAE-GAN typically involves the incorporation of a VAE encoder, a VAE decoder, and a GAN discriminator, as shown in Fig.2.7. The encoder compresses the input data into a lower-dimensional latent space representation, while the decoder produces the reconstruction from this latent space. The discriminator, in turn, distinguishes between real and generated images, providing feedback to both the encoder and the decoder to improve the quality of generated samples.

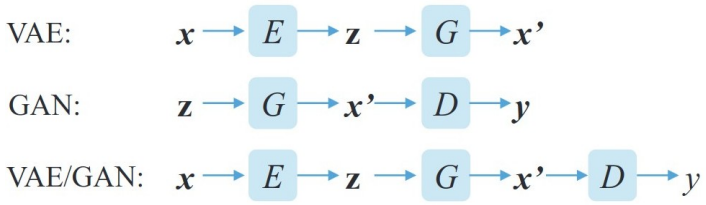


Fig. 2.7 The architectures of VAE, GAN, and VAE-GAN. Drawn from [16]

The integration of VAEs and GANs in VAE-GAN and its variants opens up numerous possibilities in the field of image synthesis. By leveraging the encoding capabilities of VAEs and the image generation abilities of GANs, researchers aim to produce high-quality, diverse, and controlled image generation models. Ongoing research continues to refine and optimize these architectures, loss functions, and training procedures to unlock the full potential of VAE-GAN and its applications in various domains.

As the field of diffusion models continues to advance, researchers are exploring the integration of diffusion models with other generative models to enhance their capabilities. One such integration is exemplified by DiffuseVAE [176]. DiffuseVAE combines the variational autoencoder (VAE) model with the diffusion model, conditioning the diffusion model with the reconstruction output from VAE. By incorporating the encoding capability of the VAE with the diffusion model, DiffuseVAE enables the generation process to be controlled and improves the overall quality of generated samples.

Another integration is demonstrated in Diffusion-GAN [260]. In this approach, a discriminator is incorporated into the diffusion process to distinguish between real and fake noisy images at each diffusion step. By utilizing the discriminator, Diffusion-

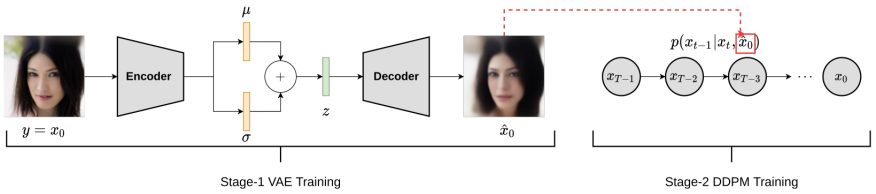


Fig. 2.8 The architectures of diffuseVAE. Image is taken from [176].

GAN enhances the realism of the generated images and produces more realistic results compared to traditional diffusion models.

These integrations of diffusion models with other generative models showcase the potential to combine different techniques and frameworks to achieve improved generation quality, controllability, and realism in the generated samples.

2.2 Disentanglement Representation

When generating faces, a common problem arises - what kind of face should the model produce? Should it have white skin, a small nose, or any other specific feature? To address this challenge, a disentangled representation [18] can be employed, which involves separating the different facial features, such as eye color, nose shape, and lip size. By disentangling these features, new images can be generated by combining different combinations of these features.

Disentangling the factors of variation within data is a critical challenge in the fields of machine learning and computer vision. Typically, complex data is represented using a feature space. Disentangled representation involves encoding each observable feature of the data separately within this feature space, so that each element of the space carries an interpretable semantic meaning. Specifically, when a feature of the data, such as the color of a face, changes, the corresponding element in the feature space should also change. This process aims to separate the underlying factors of variation within a dataset or system, allowing them to be analyzed and examined independently. Achieving disentangled representation is an important goal because it can enhance interpretability, controllability, generalizability, and robustness of the model.

2.2.1 Traditional Statistical Approaches

In order to achieve Disentangled Representation Learning (DRL) without relying on deep learning techniques, there are several traditional methods that have been proven effective. One of the most representative algorithms in this regard is Principal Component Analysis (PCA). PCA is widely recognized for its ability to disentangle latent factors and has been successfully utilized in various applications.

Principle Component Analysis (PCA) Principle Component Analysis (PCA) [262] is a widely used dimensionality reduction technique in machine learning that involves projecting high-dimensional data onto a lower-dimensional subspace while retaining as much of the original variance as possible. PCA applies a linear transformation to identify the underlying structure or patterns in the data by projecting it onto a new set of orthogonal axes known as principal components. The first principal component is selected to capture the direction of maximum variance in the data, while subsequent components capture the remaining variance in orthogonal directions. While PCA and disentangled representation learning have distinct objectives and methodologies, they share a common goal of identifying and isolating important features or factors of variation in the data. In particular, the principal components identified by PCA can be interpreted as the most important factors of variation in the data, and hence, can be viewed as a form of disentangled representation.

However since it is a linear technique, this can limit its effectiveness in capturing complex features that may exist within the data. Finally, PCA can be sensitive to outliers, as the presence of outliers can significantly affect the results of the analysis.

Human Face Modeling The concept of disentangled representation is also employed in the parametric modeling of 3D faces. In this context, disentanglement is accomplished by applying PCA on different groups of the dataset. BFM (Basel Face Model)[179] is such a model whose input consists of the shape, expression and texture parameters. To learn the shape variation, PCA is performed on a large dataset of diverse subject. The expression component of the BFM was obtained using a similar approach, but using a dataset of faces with a range of expressions. These components capture the deformation of the face caused by expression and shape changes, allowing the generation of realistic 3D models of faces with various expressions and shapes. Additionally, the texture model of BFM is constructed by performing PCA on a set of facial texture maps.

Human Body Modeling Similar idea is also applied on human body. The Skinned Multi-Person Linear (SMPL) model is a widely used method for modeling human body shape and pose. It is a statistical body model based on optimization method that can represent a wide variety of body shapes and poses using a low-dimensional parameterization. The SMPL model has been used in computer graphics, computer vision, and machine learning applications such as virtual try-on, motion capture, and pose estimation. The model provides a compact and expressive representation of the human body, which can be used for a variety of tasks.

To achieve this disentangled representation, the SMPL model uses a multilinear

model to represent the 3D human body. The pose variation is first learned by using multi-pose dataset. Then a joint regressor is used to alter the joint position of different subjects. Finally, the shape variation is learned by using different subjects but with the same pose. The whole process is illustrated in Fig.2.9.

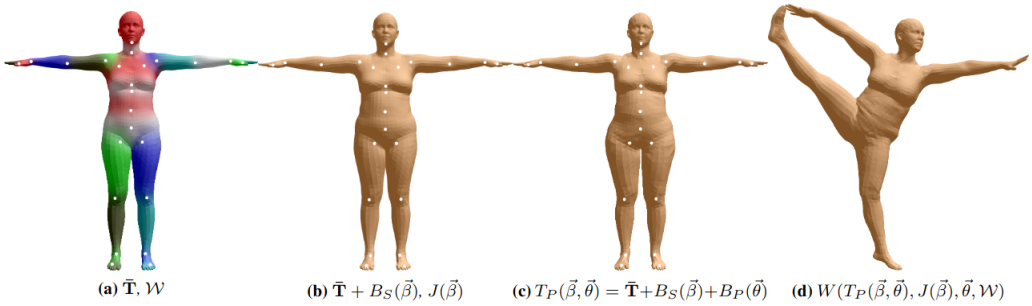


Fig. 2.9 Process of SMPL model. Image is taken from [153]. (a) The template mesh, denoted as $\overline{T}$, is shown with color-coded weights and white joints. (b)Shape-driven deformation is applied to the template mesh. The shape deformation, represented by $B_S(\vec{\beta})$, modifies the template mesh based on the shape parameter $\vec{\beta}$. Additionally, the joints, denoted as $J(\vec{\beta})$, are repositioned accordingly. (c) In addition to shape deformation, pose-dependent shape deformation is introduced. This deformation, denoted as $B_P(\vec{\theta})$, takes into account the influence of pose on the shape of the model. It further modifies the template mesh based on the pose parameters $\vec{\theta}$. (d) Vertex deformation is performed using dual quaternion skinning, resulting in the final pose of the model.

2.2.2 VAE-based Methods

Generative models, such as VAEs, have demonstrated remarkable potential in unsupervised disentangled representation learning. These models can effectively capture and separate underlying factors of variation in the data without explicit supervision. VAEs learn a low-dimensional latent representation of the data that can be used for generation and manipulation tasks. One of the most widely-used methods that learns the disentangled representation is β-VAE [86]. The objective function writes:

$$\mathbb{E}_{z\sim q(z|x)}[-\log p(x \mid z)] + \beta D_{KL}(q(z \mid x)\|p(z)). \tag{2.28}$$

When $\beta = 1$, β-VAE has the same formulation as the original VAE objective function. Increasing the value of β will encourage the disentanglement in the latent space. Intuitively, it will increase the independence among the factors of the latent space and carry less information of reconstruction. [28] propose to explain β-VAE using the information bottleneck method. The stratety is to gradually increase the information capacity of the latent channel and the objective function is written as follows:

$$\mathbb{E}_{z\sim q(z|x)}[-\log p(x \mid z)] + \gamma|D_{KL}(q(z \mid x)\|p(z)) - C|, \tag{2.29}$$

where γ and C are hyperparameters. In order to ensure a high-quality reconstruction and effective disentanglement, the value of C will gradually increase from 0 to a significant value. This gradual increment encourages the latent space to contain ample information that is necessary for achieving excellent reconstruction quality while also ensuring a strong disentanglement capability.

FactorVAE [118] further augments Eq.2.28 by adding a Total Correlation (TC) term $-\gamma D_{KL}(q(z)\|\bar{q}(z))$, where $\bar{q}(z) = \prod_j q(z_j)$ where $z_{(j)}$ stands for j-th component of z. This term measures the dimension-wise dependence in the latent space. [30] propose to decompose $D_{KL}\left(q\left(z \mid x\right)\|p(z)\right)$ into three parts: (i) index-code mutual information, (ii) total correlation, (iii) dimension-wise KL divergence. It writes:

$$\begin{aligned} D_{KL}\left(q(z \mid x)\|p(z)\right) = & \underbrace{D_{KL}\left(q(z,x)\|q(z)p(x)\right)}_{\text{(i) Index-code Mutual Information}} \\ & + \underbrace{D_{KL}\left(q(z)\|\prod_j q\left(z_j\right)\right)}_{\text{(ii) Total Correlation}} \\ & + \underbrace{\sum_j D_{KL}\left(q\left(z_j\right)\|p\left(z_j\right)\right)}_{\text{(iii) Dimension-wise K L Divergence}} . \end{aligned} \tag{2.30}$$

where index-code mutual information, as introduced in [28], will encourage compact and disentangled representation, then total correlation forces the model to find statistically independent factors in the data distribution, finally dimension-wise KL Divergence prevents the latent space to be far away from the prior. [30] validate the significance of the Total Correlation (TC) term in the decomposition of disentangled representation learning. They demonstrate that by penalizing this term, it is possible to effectively learn disentangled representations. Inspired by the β-VAE framework, where a hyperparameter is introduced to adjust the importance of each term in the decomposition, they propose a variant called β-TCVAE. The results are shown in Fig.2.10.

The aforementioned VAE-based methods are unsupervised. It is interesting to note that we can learn the disentangled representation using VAE in a (semi)supervised manner. [122] propose to use semi-supervised training to learn a representation that contains the label y. Then the inference model becomes: $q_\phi(y, z \mid x) = q_\phi(z \mid x, y)q_\phi(y \mid x)$. If the label is known , the ELBO becomes:

(a) Baldness (-6, 6) (b) Face width (0, 6) (c) Gender (-6, 6)

Fig. 2.10 Latent traversal of β-TCVAE on CelebA dataset. Face width is only manifested in one direction of a latent variable, so it only shows a one-sided traversal. Drawn from [30]

$$\log p_\theta(x, y) \geq \mathbb{E}_{q_\phi(z|x,y)} \left[\log p_\theta(x \mid y, z) - D_{KL}(q_\phi(z|x, y)||p(z)) + \log p_\theta(y)\right]. \tag{2.31}$$

The objective for supervised training is denoted as $-\mathcal{L}(x, y)$.

When the label is missing, it is treated as a latent variable, and posterior inference is performed on it. The resulting bound for handling data points with an unobserved label y is as follows:

$$\begin{aligned} \log p_\theta(x) &\geq \mathbb{E}_{q_\phi(y,z|x)} \left[\log p_\theta(x \mid y, z) - D_{KL}(q_\phi(z|x, y)||p(z)) + \log p_\theta(y)\right] \\ &= \sum_y q_\phi(y \mid x)(-\mathcal{L}(x, y)) + \mathcal{H}\left(q_\phi(y \mid x)\right). \end{aligned} \tag{2.32}$$

The objective for unsupervised training is denoted as $-\mathcal{U}(x)$ Finally, the ELBO on the whole dataset becomes:

$$\mathcal{J} = \sum_{(x,y)\sim\tilde{p}_l} \mathcal{L}(x, y) + \sum_{x\sim\tilde{p}_u} \mathcal{U}(x). \tag{2.33}$$

where $\tilde{p}_l$ and $\tilde{p}_u$ represent the distribution of labeled data and unlabeled data respectively.

Furthermore, [219] propose a framework that facilitates the learning of disentangled representations of data within the domain of VAEs. This framework leverages partially-specified graphical model structures and employs semi-supervised learning schemes. To achieve this, they introduce hybrid generative models that combine structured graphical models and unstructured random variables within the same latent space. The structured component y is intended to represent the label, while the remaining information is encoded into unstructured random variables z. For instance, in the MNIST database, the handwriting style is encoded in z, while the numerical digit is encoded in

y. The results on the MNIST database is presented in Fig.2.11.

Fig. 2.11 Illustrations of results from supervised disentangled representation learning. The images in the leftmost column serve as reference styles, assumed to be encoded in z. The remaining images are generated using the style z and numerical labels y ranging from 1 to 9. Image is taken from [219].

However, recognizing that a single label y may not fully capture the features associated with it, [111] propose a different approach called CCVAE (characteristic capturing VAE). Instead of directly incorporating the label value into the latent space, the goal is to learn a representation of the label y that can effectively affect the associated features. The graphical model depicting this concept is illustrated in Figure 2.12. To establish the relationship between the latent variable z_c^i and the label y^i, where i represents the i-th label, two distributions are introduced: $q_\varphi(y^i \mid z_c^i)$ and $p_\psi(z_c^i \mid y^i)$. Here, z_c^i represents a subset of the latent variable z, which is calculated from the input data x using an inference model $q_\phi(z|x)$. The reconstruction process is performed using a generative model $p_\theta(x \mid z)$. By connecting the label y^i and the latent space subset z_c^i, it becomes possible to modify the features associated with the label y^i by manipulating z_c^i. This approach allows for the alteration of specific characteristics related to the label, enabling more nuanced control over the generated output.

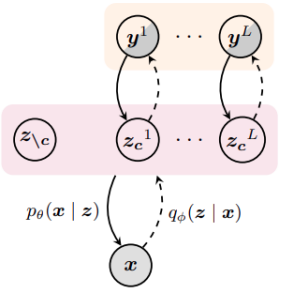


Fig. 2.12 CCVAE graphical model. Drawn from [111]

In our research [300], we also focus on learning a representation of y that enables us to manipulate its characteristics with fine-grained control. We extend this work by incorporating a discrete latent space and employing a two-step learning procedure as proposed by [302]. This approach enhances our ability to accurately manipulate features and generate high-quality image reconstructions. For more comprehensive information on these two studies, please refer to Chapter 5 and Chapter 6.

2.2.3 GAN-based model

The GAN framework offers an alternative generative model for learning unsupervised disentangled representations. Similar to VAEs, additional terms can be incorporated into the loss function or a prior can be introduced. A seminal work in this area that utilizes GAN to learn disentangled representations is InfoGAN [32].

The input is composed of two elements: (i) an unstructured noise vector represented as z, and (ii) a discrete latent code c that is sampled from a mixture of one-hot variables, it is specifically designed to capture independent features or variations present in the data distribution. To enable c to disentangle specific factors in the data, InfoGAN introduces a variational regularizer in the form of mutual information, denoted as $I(c, G(z, c))$, weighted by a hyperparameter λ. Consequently, the objective of InfoGAN can be reformulated as follows:

$$\min_G \max_D V_I(D, G) = V'(D, G) - \lambda I(c; G(z, c)), \tag{2.34}$$

where $V'(D, G)$ is the original objective of GAN (Eq. 2.23). However, since the posterior $p(c|x)$ is intractable, $I(c; G(z, c))$ is difficult to optimize. InfoGAN chooses to derive a lower bound of it by approximating $p(c|x)$ with a auxiliary distribution $q(c|x)$:

$$\begin{aligned} I(c; G(z, c)) &= H(c) - H(c \mid G(z, c)) \\ &= \mathbb{E}_{x \sim G(z,c)} \left[\mathbb{E}_{c' \sim p(c|x)} \left[\log p\left(c' \mid x\right) \right] \right] + H(c) \\ &= \mathbb{E}_{x \sim G(z,c)} [\underbrace{D_{\mathrm{KL}}(p(\cdot \mid x) \| q(\cdot \mid x))}_{\geq 0} + \mathbb{E}_{c' \sim p(c|x)} \left[\log q\left(c' \mid x\right) \right]] + H(c) \\ &\geq \mathbb{E}_{x \sim G(z,c)} \left[\mathbb{E}_{c' \sim p(c|x)} \left[\log q\left(c' \mid x\right) \right] \right] + H(c), \end{aligned} \tag{2.35}$$

where $H(.)$ represents the entropy of random variable. The approximated posterior q can have learnable parameters so that it can be implemented as a neural network which often shares the parameters with the discriminator. The overview of InfoGAN is presented in Fig. 2.13.

ClusterGAN [164] extends the work of InfoGAN and introduces several key enhancements. As in infoGAN, the latent space consists of two components: a discrete variable c sampled from a mixture of one-hot variables and a continuous variable z sampled from a normal distribution. However, unlike in infoGAN where only the discrete variable c is projected back, the generated data is, in this approach, accurately mapped back to the entire latent space using an inverse network. The model is trained jointly with a cluster-specific loss, which encourages the generated samples to align with the desired clusters.

Information Bottleneck GAN(IB-GAN, [107]) further improved the InfoGAN by

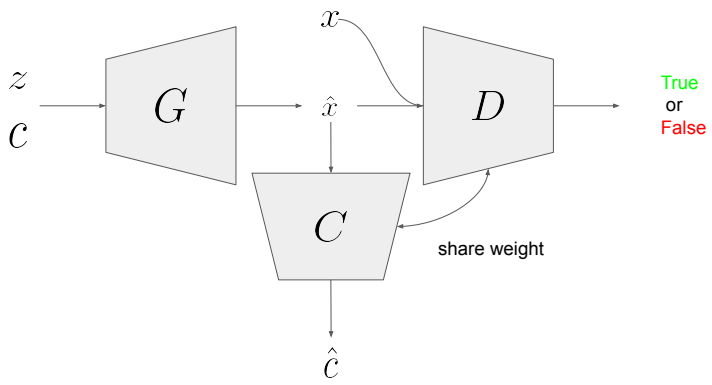


Fig. 2.13 The architecture of infoGAN.

introducing a intermediate stochastic layer e_ψ which can be considered as an encoder and project the latent space z to another representation r. Next, a weak prior $m(r)$ is established for the variable r, which follows a Gaussian distribution. The encoder is trained to minimize the KL divergence between the output r and the Gaussian prior. This process actually is one of the application of information bottleneck. Similarly to infoGAN, in addition to the discriminator, an inverse neural network is employed to project the generated output $\hat{z}$ back into the latent space z. This projection aims to maximize the mutual information between z and $\hat{z}$. The architecture is presented in Fig.2.14.

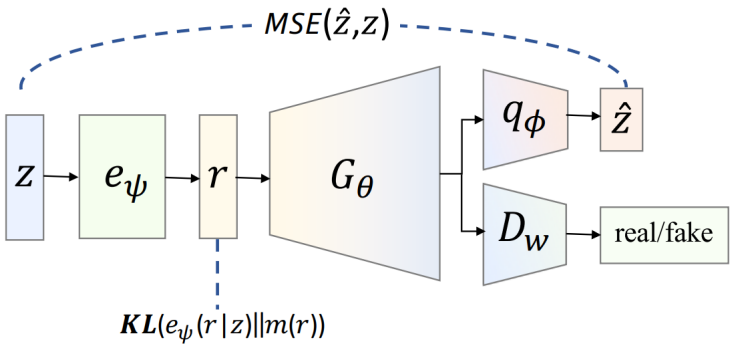


Fig. 2.14 The architecture of IB-GAN. Drawn from IB-GAN [107]

InfoGAN-CR [144] is a variant of InfoGAN that introduces a contrastive regularizer (CR) term to achieve disentanglement in a self-supervised manner. The underlying assumption behind this approach is that traversing the latent space should result in distinct changes in the generated images. The method proceeds as follows: First, several

images are generated using the model. Then, one of the latent dimensions is fixed while the remaining dimensions are randomly sampled. Subsequently, a classifier is trained to determine which specific latent variable was fixed during the generation process. This classifier acts as a contrastive regularizer, encouraging the model to disentangle the latent variables. By incorporating the contrastive regularizer, InfoGAN-CR enhances the disentanglement capability of the generative model. It encourages the model to learn representations in which each dimension of the latent space corresponds to a distinct and interpretable factor in the generated images.

In addition to unsupervised methods, there are numerous works that introduce supervision in disentangled representation learning. Similar to the VAE approach, the latent space is divided into two parts: the supervised part encodes the information relevant to the supervision task, while the second part encodes the remaining information.

[240] proposed DRGAN, a variant of the adversarial autoencoder that focuses on supervised learning. This method takes an image as input, extracting identity information from the latent space. It combines this identity information with positional information about facial features to generate desired images. Subsequently, the discriminator takes both the ground truth image and the generated image as inputs to assess and determine the pose and identification aspects. This model enables us to generate images based on specific settings in the latent space, such as pose and identity features. The architecture is shown in Fig.2.15

ELEGANT [265] and DNAGAN [264] that are adversarial autoencoders were specifically designed to encode human face attributes into its latent space. To achieve this, the model requires a pair of labeled images that have opposite attributes (such as with or without eyeglasses) as input. It then swaps latent units that are intended to represent the features associated with the given attributes. This process generates four different images: two reconstructions of the input image and two generated images of the input with the opposite attribute. The reconstructed images are used to compute the reconstruction loss, while the generated images, along with their labels, are fed into the discriminator to ensure that the latent units align well with the corresponding attributes. The architecure of this idea is shown in Fig.2.16.

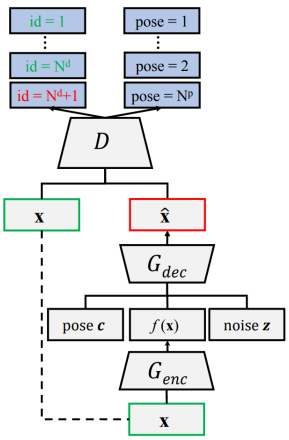


Fig. 2.15 DRGAN architecture. Drawn from [240]

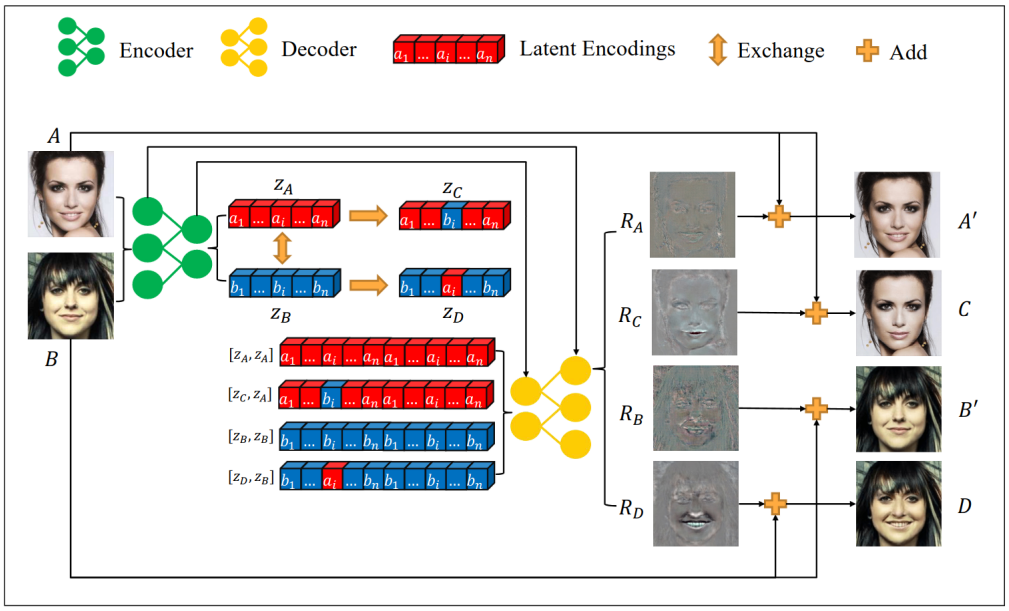


Fig. 2.16 The ELEGANT architecture. The pair of images consists of two representations, Z_A and Z_B, where each image corresponds to the opposite label of the same feature (e.g., one image represents smiling, while the other represents not smiling). The latent unit swapping aims to achieve attribute swapping aligned with the target label. Drawn from [265]

2.2.4 Diffusion Model Based Methods

As the diffusion model becomes increasingly popular, people are also beginning to explore its potential for disentangled representation learning. In a traditional autoencoder, the latent space is often a fixed-size vector that can be thought of as a compressed representation of the data. However, in a diffusion model, there is no simple way to extract meaningful data representations from the latent representation.

In traditional disentangled representation, the image is reconstructed using the decoder while its features are controlled by the representation. However, achieving it in diffusion models is challenging. Instead of directly controlling the reconstruction, a recent work by [274] proposed a novel approach. They manipulate the gradient field during the sampling process of a pre-trained diffusion model to achieve disentanglement. The encoder takes an original image x_0 as input and generates a series of latent vectors, denoted as $z_c = z_1, z_2, ..., z_c$, where c is the number of underlying factors, each representing one of the attributes in the image space. The decoder takes one of the latent representations, as well as the latent representation of image at each step x_t (from the UNet of the pre-trained diffusion model) as input, and outputs the gradient field of the corresponding attribute. With the help of the gradient, one can sample the image under the corresponding condition. To conclude, the disentangled representation in this ap-

proach is used to obtain the gradient field to guide the reverse process. The architecture is shown in Fig. 2.17

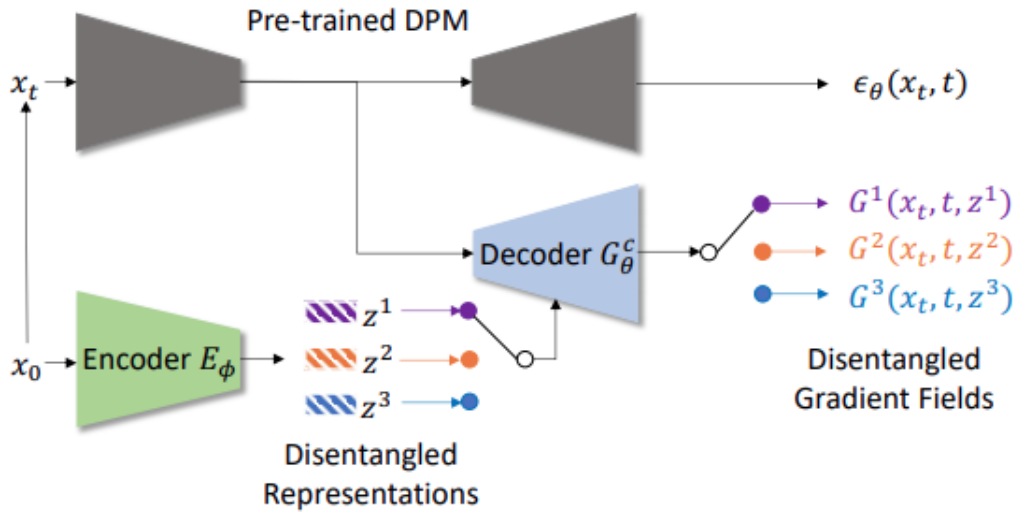


Fig. 2.17 Illustration of DisDiff. Drawn from [274]

2.2.5 Evaluation Metrics

Evaluating the quality and effectiveness of disentangled representations is an important aspect in the field of machine learning. Several evaluation metrics have been proposed to assess the disentanglement of learned representations.

For unsupervised disentangled reoresentation learning, one commonly used metric is the Mutual Information Gap (MIG) [30], which measures the degree to which each learned factor of variation is captured by a single latent dimension. A higher MIG score indicates a better disentanglement of factors. The joint distribution of a latent variable z_j and a ground truth factor v_k can be defined as $q\left(z_j, v_k\right) = \sum_{n=1}^{N} p\left(v_k\right) p\left(n \mid v_k\right) q\left(z_j \mid n\right)$, where $p\left(n \mid v_k\right)$ is the generative process for factor v_k. Assuming that the underlying factor $p(v_k)$ is known during the sampling process $p\left(n \mid v_k\right)$, the mutual information can be written as:

$$I_n\left(z_j ; v_k\right) = \mathbb{E}_{q\left(z_j, v_k\right)}\left[\log \sum_{n \in \mathcal{X}_{v_k}} q\left(z_j \mid n\right) p\left(n \mid v_k\right)\right] + H\left(z_j\right), \tag{2.36}$$

where $\mathcal{X}_{v_k}$ is the support of $p\left(n \mid v_k\right)$.

Then the MIG is defined as:

$$\frac{1}{K} \sum_{k=1}^{K} \frac{1}{H\left(v_k\right)}\left(I_n\left(z_{j^{(k)}} ; v_k\right) - \max_{j \neq j^{(k)}} I_n\left(z_j ; v_k\right)\right), \tag{2.37}$$

where $H(v_k) = \mathbb{E}_{p(v_k)}[-\log p(v_k)]$, $j^{(k)} = \text{argmax}_j I_n(z_j; v_k)_1$. K is the number of known factors.

In supervised disentangled representation learning, the classification error is a commonly used metric [219, 265]. By leveraging a pretrained classifier, one can evaluate the generated results by comparing them to the target label. This involves assessing whether the generated samples are correctly classified into their respective categories. The classification error provides a quantitative measure of the accuracy of the generated samples in terms of their assigned labels. Likewise, [111] introduces the concept of calculating the log-probability output from a classifier to verify if the feature is accurately captured by the latent space. The authors perform attribute swapping experiments, such as transferring eyeglasses from one image to another, and then compare the log-probability between the original image and the generated image. The underlying principle is that an ideal attribute swapping operation should not result in a significant alteration of the probability output from the classifier. By examining the log-probability values, one can assess the extent to which the latent space successfully represents and preserves the important features relevant to the classifier's decision-making process.

To evaluate the image generation quality, a widely used evaluation metric is called Fréchet Inception Distance (FID) [85]. The FID metric measures the similarity between the generated images and real images based on the feature representations learned by an Inception network. It uses a two-step process to calculate the distance: first step, a pretrained neural network (usually inceptionNetv3 [232]) is used to extract feature representations from both the real and generated images. The FID computes the Fréchet distance between the multivariate Gaussian distributions of the real and generated image features:

$$FID = \|\mu_r - \mu_g\|^2 + T_r\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r\Sigma_g\right)^{1/2}\right), \tag{2.38}$$

where $\mathcal{N}(\mu_r, \Sigma_r)$ and $\mathcal{N}(\mu_g, \Sigma_g)$ are multivariate Gaussian distributions of the real and generated image features respectively.

2.3 Conditional Generation

In the earlier section, we explored how disentangled representations can be employed for feature control and conditional generation. However, conditional generation can also be achieved without relying on disentanglement. Instead of using labels directly, various approaches leverage different modalities such as images or text to control the generation process. One approach involves using images to control image generation, enabling tasks like image translation. Another approach involves using text to control image features, facilitating text-to-image generation. Additionally, there are methods

that utilize images to control text generation, which is commonly known as image captioning. In this section, we will delve into these diverse types of conditional generation and highlight some of the cutting-edge works in these areas.

2.3.1 Condition on the label

Data labels are one of the simplest and most widely used forms of conditional information in machine learning. By providing a label for each data point, we can train models to learn the underlying patterns and relationships in the data, and make predictions or generate new examples based on these learned patterns.

In Sec.2.2, we have already discussed the approach of incorporating labels for supervised disentangled representation learning (DRL) which also enables generation from a label variable. A key characteristic of DRL is that the remaining information, excluding the label, is encoded into a latent space denoted as z, which is concatenated with the label variable y. Together, they constitute a complete latent space that is fed into the decoder for generation. Conditional generation does not impose any requirements on the form of the conditions or whether the remaining information is entangled. It simply refers to the process of generating data based on given conditions, without specifying how the conditions or remaining information are encoded or represented. Based on this, the supervised DRL can also be seen as a specific form of conditional generation.

As depicted in Figure 2.18, the neural network requires a mechanism to incorporate the conditional information. The architecture of the neural network can be adapted to accommodate different forms of label conditions. One straightforward approach is to concatenate the label y with the intermediate output of the model. In the work on conditional human motion generation [182], learnable tokens are employed. These tokens, with the same size as the latent representation, are added as offsets to the latent space. This incorporation of tokens enables conditional generation by encoding specific labels. On the other hand, in the work on image translation [265], the label value is directly used to generate an array with the same shape as the input image. Subsequently, this array is concatenated with the image along the depth channel, allowing for conditional discriminator based on the given label. Plug-and-Play Generative Networks (PPGN) [167] utilizes a pretrained classifier and an optimization method to align the generated data with the classifier's output based on the given label.

One approach for conditional generation based on label variable is conditional GAN (CGAN) proposed by [161], which introduces a conditional adversarial learning framework. In this framework, the generator takes both a random latent vector z and a discrete label vector y (one-hot encoding) as input. Furthermore, the discriminator network takes both the generated image and the label y as input, ensuring that the generated image corresponds to the given label. By jointly training the generator and discriminator,

the model learns to generate images that are not only coherent and realistic but also aligned with the specified label. The loss is presented as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x,y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z,y)))]. \quad (2.39)$$

Once the model is trained, modifying the label vector y will prompt the generator to generate images that are associated with the updated label. This capability enables the generation of corresponding images for different labels, showcasing the disentangled representation learned by the model.

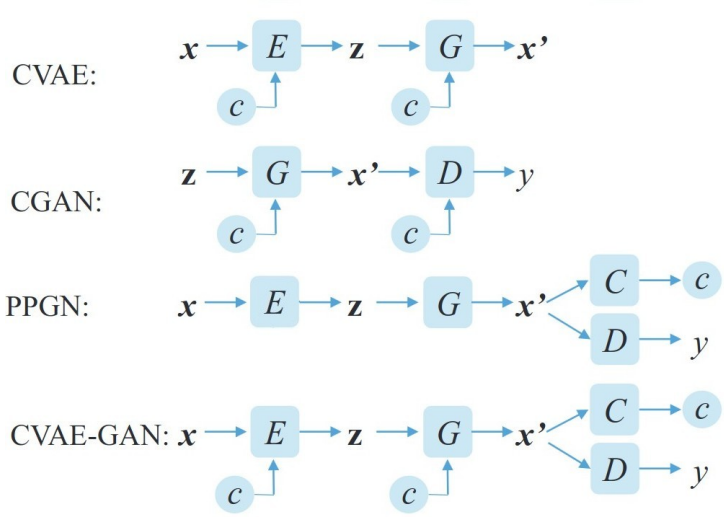


Fig. 2.18 The architectures of CVAE, CGAN, PPGN, CVAE-GAN. Drawn from [16]

Conditional VAE (CVAE) [225] extends the concept of Variational Autoencoder (VAE) by introducing stochastic neural networks for structured output prediction. They propose a conditional deep generative model with Gaussian latent variables. Notably, subsequent methods utilize the cVAE approach, employing labels to condition both the encoder and decoder as shown in Fig.2.18 (CVAE) [182, 34]. In this scenario, the approach closely resembles the method presented in [122] for the supervised case.

Additionally, techniques like Plug-and-Play Generative Networks (PPGN) [167], shown in Fig.2.18 (PPGN), aim to generate realistic images that align with specific conditions by leveraging auxiliary models and an optimization method. The key idea behind PPGN is to combine the power of generative autoencoders with pretrained conditional models such as classifiers or image caption models. The framework takes advantage of the pretrained models' knowledge about specific conditions or attributes and optimizes the generated images to align with those conditions.

Furthermore, researchers have investigated the incorporation of conditioning in the VAE-GAN architecture for generating images based on specific attributes or classes.

One such approach is the Conditional VAE-GAN (CVAE-GAN) proposed by [16]. The CVAE-GAN builds upon the VAE-GAN framework by incorporating an adversarial loss to enhance the quality of generated images. In addition to the discriminator, the CVAE-GAN utilizes a classifier to assess the consistency of the generated output with the given label, enabling more accurate generation. The architecture of CVAE-GAN is depicted in Figure 2.18.

A diffusion model can also be trained in a conditional manner. To incorporate label information for noise estimation, a frequently employed method involves training a label embedding and combining it with the time step embedding. This combined information is then fed into the noise approximator, which is responsible for generating the conditional noise samples [139].

[91] introduced a classifier-free guidance approach for the reverse process of the diffusion model to improve the conditional diffusion model. It proposes to jointly train a conditional and an unconditional diffusion model. During training, a null label $\emptyset$ is introduced and replaces the actual label y with a fixed probability, enabling unconditional training. During the sampling process, the authors combine the scores from both the conditional and unconditional distributions to obtain the final noise estimation. This combination can be calculated using the following equation:

$$\hat{\varepsilon}_\theta\left(x_t \mid y\right) = (1+s) \cdot \epsilon_\theta\left(x_t \mid y\right) - s \cdot \epsilon_\theta\left(x_t \mid \emptyset\right), \tag{2.40}$$

where $\hat{\varepsilon}_\theta\left(x_t \mid y\right)$ represents the final noise estimation for x_t conditioned on label y. The term $\epsilon_\theta\left(x_t \mid y\right)$ denotes the estimation of conditional noise, while $\epsilon_\theta\left(x_t \mid \emptyset\right)$ represents the estimation of unconditional noise. Additionally, the scale factor s serves a similar purpose as in the classifier-guided method.

One approach commonly used in diffusion models for conditioning generation involves classifier-guided sampling [231, 46]. The diffusion model can be trained unconditionally. Once the diffusion model is established, the reverse process can be guided by a classifier. Specifically, we can train a classifier, denoted as $p(y|x_t, t)$, on the noisy image x_t at time step t. The predicted mean $\mu_\theta(x_t)$ is then perturbed by the gradient of the classifier, resulting in the final predicted $\hat{\mu}_\theta\left(x_t \mid y\right)$ conditioned on y. This can be calculated using the following equation:

$$\hat{\mu}_\theta\left(x_t \mid y\right) = \mu_\theta\left(x_t\right) + s \cdot \sigma_t^2 \nabla_{x_t} \log p_\phi\left(y \mid x_t\right), \tag{2.41}$$

where s represents a scaling factor and σ denotes the diffusion model's noise level. The term $\nabla_{x_t} \log p_\phi\left(y \mid x_t\right)$ represents the gradient of the classifier with respect to x_t, which is used to perturb the predicted mean $\mu_\theta(x_t)$. According to the findings in [46], increasing the value of s comes at the cost of reduced diversity, but it can also lead to

improved image quality.

2.3.2 Image-to-image translation

There are various forms of conditioning beyond the conventional label based on human annotations. One such approach is to use images as input conditions to enhance or modify their appearance, a process commonly known as image translation. Image-to-image translation refers to the process of transferring an image from a source domain to a target domain while preserving the image content. Image-to-image translation has a wide range of applications, including style transfer, image synthesis, segmentation, restoration, and pose estimation. In this section, we will introduce popular applications of image translation and their state-of-the-art methods, such as style transfer, image inpainting, and edge-map to image. These applications enable the generation of unique and previously unseen images with distinct visual characteristics.

Fig. 2.19 Style transfer examples taken from [110].

Style transfer Style transfer is a technique that can be used to transform the style of an image, by applying the style of one image to the content of another image. It is achieved by separating the content and style of an image and then combining them in a new way.

The concept of style transfer was initially proposed by [66]. They presented a deep convolutional neural network based approach to separate the content and style of an image, and transfer the style of one image onto the content of another image. To extract the style and content features, they utilized a pre-trained VGG model [221]. The style representation was computed by calculating the Gram matrix $G^l \in \mathcal{R}^{C_l \times C_l}$ from the features of each convolutional layer, where C_l is the channel dimension of layer l. Then it can be written as follows.

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}, \tag{2.42}$$

where, F^l_{ik} represents output of the activation of the i-th filter at position k in layer l. The content was represented by the feature maps output from the fourth convolutional

layer. The optimization process was then performed on a noise-initialized image by minimizing the difference between the features of the style image and the generated image, while preserving the content of the content image.

Traditional style transfer algorithms mainly rely on optimization methods that compare the features of the generated image and the source image, which require a large number of iteration steps. Based on the work of [66], [110] proposes a perceptual loss to train an autoencoder which enables a real time style transfer. As in [66], the perceptual loss also includes two main components: the content loss, which calculates the feature distance between the content image and the generated image at the third layer of a pre-trained VGG network, and the style loss, which measures the similarity between the gram matrix of the feature outputs from each layer and that of the style image. Consequently, during training, the loss function is similar to the optimization objective of [66]. Some examples of results are shown in Fig.2.19

Another popular method for enabling style transfer is the Adaptive Instance Normalization (AdaIN) proposed by [99]. This method uses data normalization in neural networks to connect the content feature and the style feature by transferring the feature statistics. The content feature space f_c is first normalized, then scaled by the variance $\sigma(f_s)$ and shifted with the mean $\mu(f_s)$ of the style feature f_s. It can be written as follows.

$$\mathrm{AdaIN}(f_c, f_s) = \sigma(f_s)\left(\frac{f_c - \mu(f_c)}{\sigma(f_c)}\right) + \mu(f_s). \tag{2.43}$$

In order to achieve a real-time style transfer, they designed an autoencoder that computes the feature space for both the style image and content image. An AdaIN layer is then used to transfer the statistical properties of the style feature to the content feature. The desired image is obtained by passing the result through a decoder. Similar to the approach in [66], both the style loss and the content loss are utilized to optimize the neural network.

Similarly, [140] use the first few layers of pre-trained VGG as the encoder and train the corresponding decoder. They found that the whitening transformation can remove the style related information and preserve the structure of content. Thus the latent representation can be manipulated through a combination of whitening and coloring transformations (WCT).

GAN-based method Although style transfer is an interesting technique that can generate many incredible images, its application scenarios are quite limited as it relies mainly on neural networks' ability to extract and compare image features. Next, we will introduce several GAN-based method to achieve two domains translation, which can not

only achieve style transfer but also have much broader application scenarios, such as segmentation, edge-to-image, sketch-to-image, etc.

[105] proposes pix2pix to leverage the conditional GAN to solve the supervised image-to-image problem. The training process consists of two parts. (i) The generator receives the condition as input and generates images that aim to deceive the discriminator into classifying them as real. In addition to fooling the discriminator, the generator's loss function incorporates the L1 distance to minimize the dissimilarity between the generated images and the ground truth images in the target domain. (ii) The discriminator takes both the condition and images as input and is trained to differentiate between real images and those generated by the generator. Its objective is to accurately classify the authenticity of the input images based on the given condition. The condition can be replaced by any other auxiliary information such as a sketch, a semantic segmentation map, or a black-and-white image. This work has been extended to many works, such as [251, 5, 255], etc.

While Pix2Pix is a highly effective framework for image-to-image translation, it relies on annotated images for training, which can be expensive to obtain. CycleGAN [296] is also a popular framework for image translation, which does not require paired data during training. It consists of two mapping functions, $G_{X\to Y}$ and $G_{Y\to X}$, that allow domain conversion between X and Y. Two discriminators, D_X and D_Y, are used to differentiate between the generated images and real images in each domain. In addition to the GAN loss, a cycle consistency loss is also employed to ensure consistency in image content. The concept behind this loss is that if an image x from domain X is transferred to domain Y and then back to the original domain, the final image should be similar to the original input: $x \approx G_{Y\to X}(G_{X\to Y}(x))$. The results are shown in Fig.2.20. There have been many variations of this idea developed, such as [297, 8].

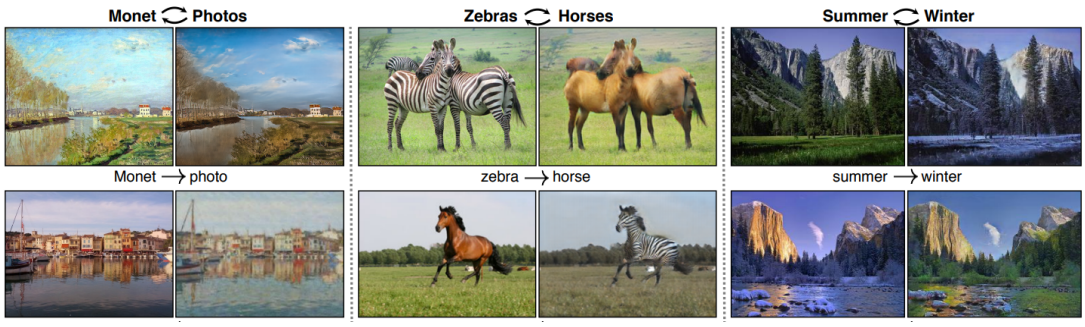


Fig. 2.20 Image-to-image translation results of cycleGAN [296].

GAN inversion An alternative and widely used approach for image translation is to convert a given image x into noise in the latent space z of a GAN. Once the image is transformed into noise, it can be manipulated to generate the desired output image.

This technique was initially introduced by [295], where they proposed to find the initial noise z in the GAN's latent space by solving the following equation:

$$z^* = \arg\min_z \ell(G(z), x). \tag{2.44}$$

This process is referred to as *inversion* and was named as such in the work of [41]. There are two categories of methods used to solve Eq.2.44: learning-based methods [51, 54, 294] and optimization-based methods [295, 1, 204]. In a similar fashion to early-stage style transfer techniques, optimization-based approaches involve the iterative optimization of input noise to generate high-quality reconstructed images. These methods tend to produce superior reconstruction results, but they come with a high computational cost due to the iterative nature of the optimization process. On the other hand, learning-based methods provide a computationally efficient alternative, but they typically yield lower-quality reconstructions compared to optimization-based methods.

The early work by [181] focused on inverting conditional GANs (icGAN) through the training of two encoders: one for the latent space z and another for the condition y. This approach enabled the extraction of the latent space z from an image, allowing for feature manipulation by modifying the label y. The architecture of icGAN is shown in Fig.2.21.

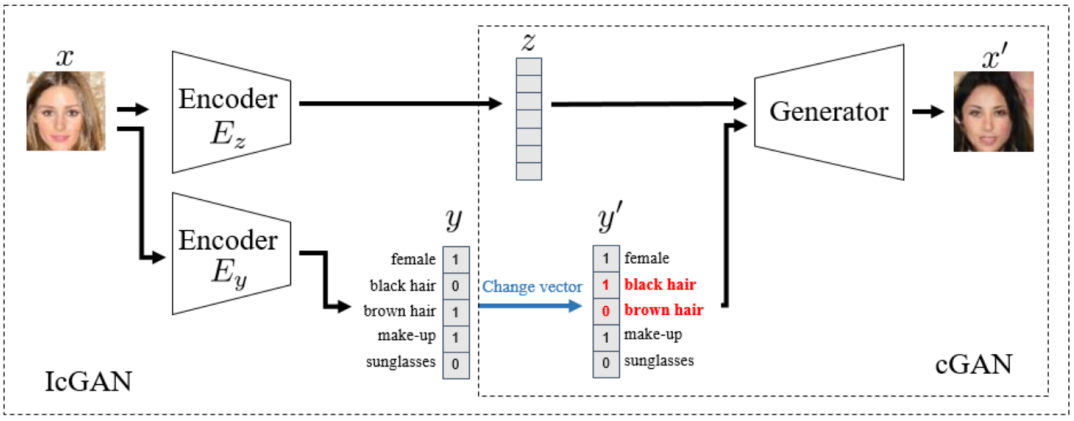


Fig. 2.21 Architecture of icGAN. Drawn from [181].

The popularity of StyleGAN [115], as mentioned in Section 2.1.3, has played a significant role in driving the development of numerous GAN inversion methods specifically designed for editing human faces. Among these methods, [204] propose a framework called pixel2style2pixel (pSp) for image-to-image translation. The pSp framework employs an encoder that learns the $\mathcal{W}+$ space of StyleGAN. This space comprises 18 feature layers spanning from deep to shallow layers within the StyleGAN generator. The encoder can accept any form of condition as input, and its output, combined

with random noise, facilitates a wide range of image translation tasks by manipulating the $\mathcal{W}+$ space of a given image. For instance, the encoder takes a segmentation map as input and produces a representation in the $\mathcal{W}+$ space. Subsequently, a pretrained StyleGAN model can accept a mixture of this representation and a randomly sampled latent vector as input, generating an image that corresponds to the given segmentation map.The architecture of the pSp framework is illustrated in Figure 2.22, and the corresponding results are presented in Figure 2.23.

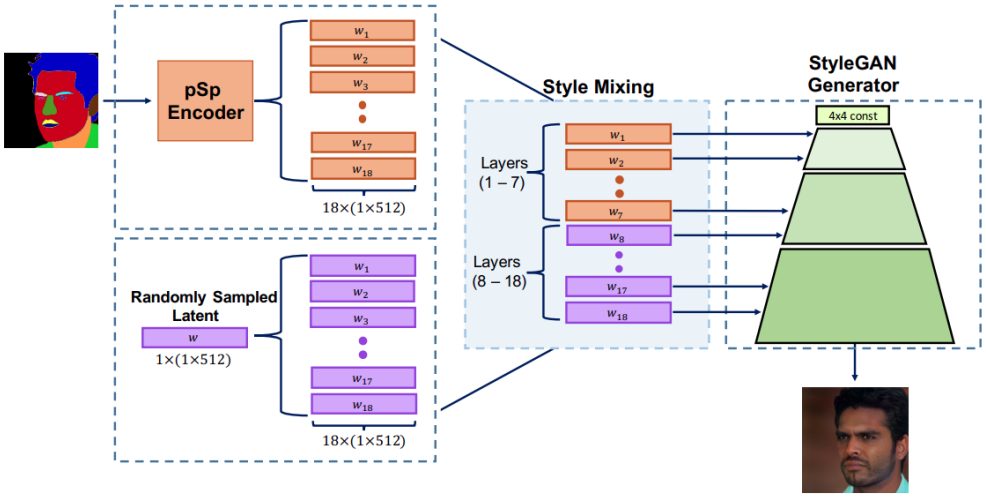


Fig. 2.22 The architecture of pSp. Drawn from [204]

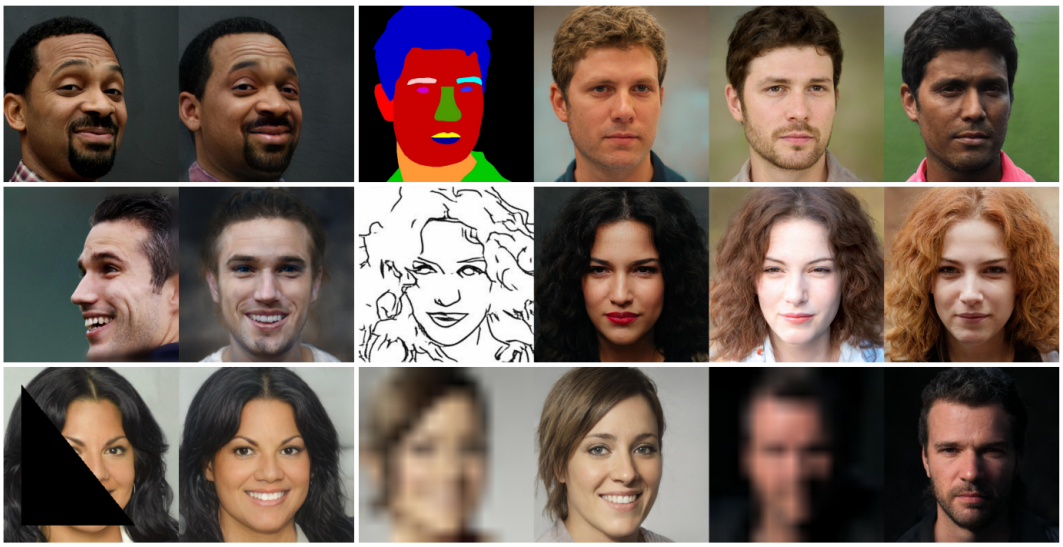


Fig. 2.23 The image-to-image translation results for human face based on StyleGAN. Images are taken from [204]

Building upon the pretrained pSp encoder, [276] employ a learned neural network

to edit the $\mathcal{W}+$ space, thereby achieving face image manipulation. This manipulation includes adding glasses, altering age, and more. Similarly, [96] propose a styleGAN encoder based on the transformer architecture [248] to learn the $\mathcal{W}+$ space. Images can be edited using a latent model, also based on the transformer, to transform the $\mathcal{W}+$ space according to desired modifications. [178] explore the utilization of text to manipulate images in conjunction with CLIP [191], a neural network-based model for evaluating the connection between texts and images. They employ a neural network to optimize or estimate the latent space, with the objective of generating an image that minimizes the CLIP loss. This approach enables text-guided image synthesis, where the desired image output is driven by textual input, resulting in visually coherent and contextually relevant image manipulations. Building upon previous research endeavors that aimed to approximate the latent space and reconstruct given images, [4] introduce a novel approach. They utilize a hypernetwork that takes both the given image and its reconstruction as input, allowing the estimation of parameter offsets for the pretrained StyleGAN. This technique enhances the ability to manipulate and generate images within the StyleGAN framework.

2.3.3 Text-to-image generation

As individuals read text, their minds often generate corresponding mental images based on the textual descriptions. With the advancements in deep learning, neural networks are now capable of emulating this process. By inputting a text prompt, a neural network can generate an image that is visually coherent with the given text. This remarkable capability enables the network to bridge the gap between textual information and visual representation, opening up new possibilities in image synthesis and understanding.

Text-to-image generation in its early stages primarily relied on the application of Generative Adversarial Networks (GANs). [200] introduced the concept of conditional GANs, where a text input serves as a condition for an image generator to produce a corresponding image. This approach allows for the synthesis of images based on specific textual descriptions. In a similar vein, attGAN [268] proposed a multi-stage generator for text-to-image synthesis. Their approach involves introducing text information at each stage of the generation process, progressively generating images from low resolution to high resolution. Additionally, each generated image is passed through a conditional discriminator, which aids in providing feedback and guidance for the generator. [133] adopt the architecture of attGAN, design a channel-wise attention module and a word-level discriminator to enhance the controllability of the generation.

With the notable success of VQVAE and its extension VQGAN [58], the representive methods start becoming more and more popular. Just like text tokenizers in natural

language processing (NLP), there is a trend in the image generation domain towards developing a series of models, including VQGAN, that aim to become image tokenizers. In NLP, tokenizers split text into semantic units such as words or subwords for further processing and analysis. Similarly, the goal of image tokenizers is to transform images into processable representations.

Based on this idea, DALL-E [195] utilizes a large-scale dataset to train a representative model. Initially, DALL-E employs a discrete Variational Autoencoder (dVAE) to compress the input image, treating it as image tokens. Subsequently, a transformer architecture is employed to transform text tokens into image tokens. The final image is reconstructed using the decoder of the dVAE, resulting in a generated image that aligns with the given textual input.

DALL-E2 [194] incorporates the CLIP text encoder as a crucial element, utilizing it to extract the underlying representation of the input text. By leveraging the CLIP text encoder, DALL-E2 aims to establish a link between the domains of text and images, with the goal of learning image representations from textual representations. This process is accomplished through the utilization of either a diffusion model or an autoregressive model. Ultimately, an image decoder is employed to generate the intended image that corresponds to the provided text description.

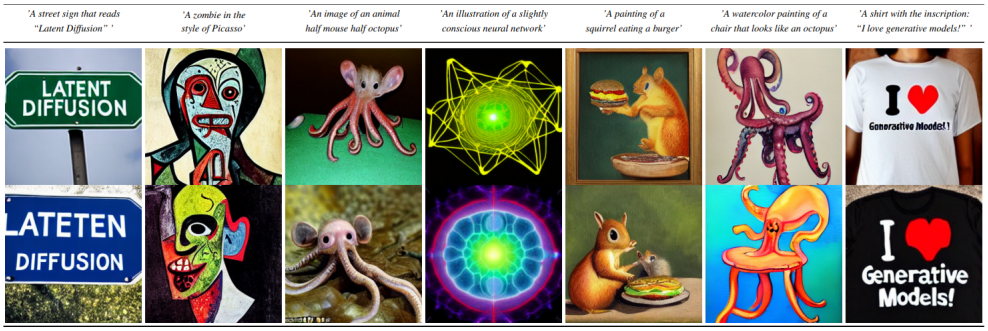


Fig. 2.24 Text-to-image results from Stable Diffusion. Drawn from [205]

Similarly, Stable diffusion [205] employs a VQGAN [58] to compress the input image. Subsequently, a diffusion model is trained to generate images, conditioned with textual information. In contrast to previous methods that establish connections between text tokens and image features, Stable diffusion integrates text guidance into each block of the noise approximator ϵ_θ through a cross-attention layer from the transformer architecture. The architecture is shown in Fig.2.25.

The combination of a powerful autoencoder and diffusion models (DM) have recently gained significant attention as the new state-of-the-art approach for text-to-image generation. The prevailing method involves compressing the image into a discrete latent space with an autoencoder, followed by training a diffusion model to learn the genera-

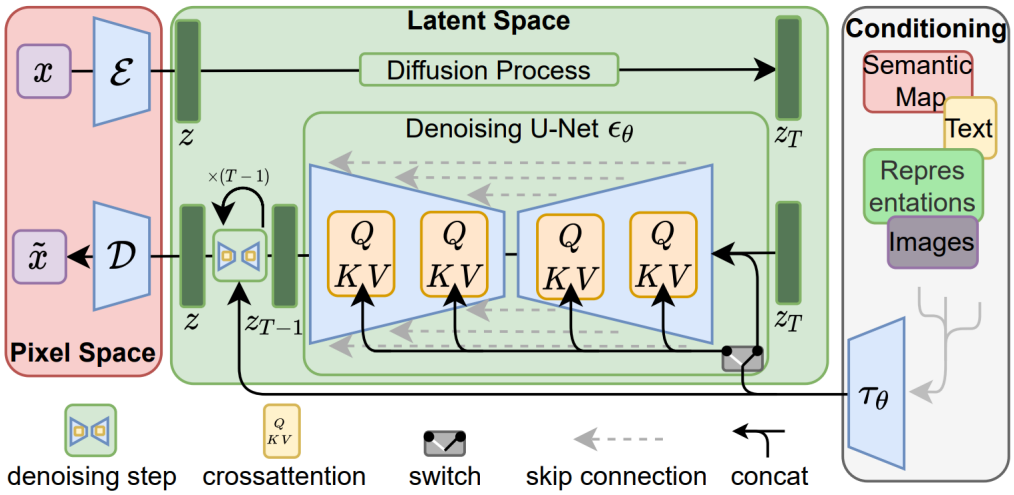


Fig. 2.25 The overview of Stable Diffusion. Drawn from [205]

tion of such a conditioned latent space. Similar to GAN, the availability of numerous powerful pretrained models opens up a plethora of possibilities for image editing without requiring training from scratch [117, 285, 63]. These methods have demonstrated significant potential in enhancing various aspects of image editing and have paved the way for further advancements in the field.

CHAPTER 3

4D Facial Expression Diffusion Model

Given the increasing popularity of the diffusion model, conditional generation has gained significant attention due to its impressive results. In this chapter, our objective is to harness the concept of the diffusion model and utilize a flexible reverse process to accomplish diverse conditional generation tasks in the realm of 4D facial expression generation.

Generating realistic facial expressions is a challenging task, mainly due to limited datasets and computational resources. There are several possibilities to conduct expression generation. (i) The regression method. Previous works ([216, 187]) have predominantly utilized recurrent neural networks (RNNs) to predict the displacements of the whole mesh. Nevertheless, performing conditional generation directly with Recurrent Neural Networks (RNNs) poses challenges. As a result, these approaches primarily concentrate on unconditional generation, where no specific conditions are imposed. Moreover, the generated outputs lack randomness, resulting in a one-to-one mapping between each input and its corresponding output. (ii) VAEs can indeed be utilized for facial animation generation and offer a convenient framework for conditional generation (cVAE). However, as mentioned in Section 2.1.1, VAEs face challenges in capturing high-frequency information, which is crucial in the context of facial animation. (iii) cGAN is also another possibility for facial animation generation. [175] utilize cGAN to generate landmarks displacement for facial animation. However, their approach primarily concentrated on pure conditional generation using labels. Additionally, their method heavily relied on complex geometry transformations and training with GANs, which introduced challenges in the training process and required intricate pre- and post-processing steps.

The primary goal of our study is to introduce a method suitable for diverse application scenarios in 4D facial expression generation. We recognize that diffusion models possess the capability to generate high-quality data and offer a flexible reverse process that can be guided by various condition sampling techniques. Therefore, we have chosen to employ diffusion models for facial animation generation.

Our work specifically focuses on various conditioning factors, including labels, text,

partial sequences, and even facial geometry. By incorporating these different conditions into the reverse process of diffusion model, we aim to enable more versatile and controllable facial expression generation. This allows us to tackle a wide range of applications and explore the potential of diffusion models in the domain of facial expression synthesis. By leveraging the flexible nature of diffusion models and incorporating diverse conditioning factors, we aim to overcome the limitations of previous approaches and provide a more robust and adaptable solution for 4D facial expression generation.

You can find hereater an article titled "4D facial diffusion model," which has been submitted to the ACM Transactions on Multimedia Computing Communications and Applications. The authors of the article are Kaifeng Zou, Sylvain Faisan, Boyang Yu, Sébastien Valette, and Hyewon Seo.

3.1 Abstract

Facial expression generation is one of the most challenging and long-sought aspects of character animation, with many interesting applications. The challenging task, traditionally having relied heavily on digital craftspersons, remains yet to be explored. In this paper, we introduce a generative framework for generating 3D facial expression sequences (i.e. 4D faces) that can be conditioned on different inputs to animate an arbitrary 3D face mesh. It is composed of two tasks: (1) Learning the generative model that is trained over a set of 3D landmark sequences, and (2) Generating 3D mesh sequences of an input facial mesh driven by the generated landmark sequences. The generative model is based on a Denoising Diffusion Probabilistic Model (DDPM), which has achieved remarkable success in generative tasks of other domains. While it can be trained unconditionally, its reverse process can still be conditioned by various condition signals. This allows us to efficiently develop several downstream tasks involving various conditional generation, by using expression labels, text, partial sequences, or simply a facial geometry. To obtain the full mesh deformation, we then develop a landmark-guided encoder-decoder to apply the geometrical deformation embedded in landmarks on a given facial mesh. Experiments show that our model has learned to generate realistic, quality expressions solely from the dataset of relatively small size, improving over the state-of-the-art methods. Videos and qualitative comparisons with other methods can be found at `https://github.com/ZOUKaifeng/4DFM`.

3.2 Introduction

3D facial expression synthesis is a fundamental, long-sought problem in face animation and recognition, with many applications. Due to the inherent subtlety and sophistication of facial expressions, as well as our sensitivity to them, the task is extremely complex.

It has traditionally relied on time- and skill-intensive design work by trained artists. The prevailing shape and motion capture technology has changed this paradigm, allowing the algorithmic reconstruction of 3D face shapes and motions of real people. At the same time, its remarkable achievements in the last decades have boosted data-driven approaches to face modeling, which have been succeeded by recent deep learning-based methods. A common strategy is to regress the 3D facial expression of a subject from a 2D video in a frame-by-frame manner[241, 60, 42, 77]. However, such *reconstructive* approach is limited to reproduce facial expressions that have been observed, and requires deformation transfer or animation retargeting to reuse the captured animation of a face to a target face. *Generative* models such as Generative adversarial nets (GANs)[69] and Variational autoencoders (VAEs)[124] can be deployed to the problem of synthesizing realistic yet controllable facial animation that are not limited to a specific observation. However, with a few exceptions[258, 216], most existing works focus on the body motion generation, with various condition signals including text, expression label, or music [182, 236, 136, 134, 76]. This is mainly due to the compact, readily available skeleton-based representation of the body[153], the relatively large set of action vocabulary, and the availability of rich 3D body motion datasets [159, 189, 184, 134, 103]. Unfortunately it is not yet the case with the 3D facial expression.

In this paper, we address the challenging problem of 3D dynamic facial expression generation, one that has not yet received a lot of attention. Most available 3D facial expression datasets [288, 196, 33, 40, 59] come in the form of dense triangular meshes containing thousands of vertices. It is computationally expensive to train a generative model directly using all the vertices. Therefore, similarly to most successful models for 3D facial animation generation, we use a set of predefined 3D face landmarks to represent the dynamics of facial motion. Typically, landmarks are located on facial features that are highly mobile during animation, such as the face outline, eyes, nose, and mouth. The specific aim of the 3D facial animation generation is to learn a model that can generate facial expressions that are realistic, appearance-preserving, rich in diversity, with various ways to condition it such as categorical expression labels. Prior works that have attempted to model the temporal dimension of the face animation [174, 175, 216, 256] mostly leverage auto-regressive approaches, such as Long short-term memory (LSTM) [93] and Gated recurrent units (GRUs) [35]. Here, we propose to use a Denoising Diffusion Probabilistic Model (DDPM) [224, 231, 89], a generative approach that has achieved remarkable success in several domains, such as image generation[205, 194, 211], audio synthesis[127], language modeling[139] and point cloud generation[156]. A DDPM has the nice property of being trainable unconditionally whereas the reverse process can still be conditioned using, a classifier-guidance [46], for instance. This allows us to define the following paradigm: a DDPM is learned

unconditionally and several downstream tasks associated with several conditional generations are developed from the same learned model, such as expression control (with label or text), expression filling (with partial sequence(s)), or geometry-adaptive generation (with facial geometry). This makes the proposed approach highly flexible and efficient, benefiting from the generative power of diffusion models while circumventing their limitations of being resource-hungry and difficult to control.

We note that, concurrent to this work, several works have also adopted diffusion models for human motion generation [237, 286, 119]. However, to the best of our knowledge, we are the first to adapt diffusion models to 3D face expression generation. More importantly, although approaches developed in [237, 286, 119] enable different forms of conditioning, they require the diffusion model to be retrained for each way of conditioning.

While the task of 3D facial animation generation has been reduced to the estimation of a temporal sequence of 3D face landmark sets, it is then necessary, in a second task, to compute a sequence of animated meshes. We use an encoder-decoder model similar to [175], which retargets the expression of a 3D face landmark set to the neutral 3D face mesh by computing its per-vertex displacement, in a frame-by-frame manner. Unlike [175], however, we take into account the different morphological shapes of the neutral mesh to adapt the estimation of per-vertex displacements. Results thus obtained validate the effectiveness of the proposed approach.

In summary, our key contributions are as follows: (1) We successfully use a DDPM to propose an original solution to the conditional generation of 3D facial animation. To the best of our knowledge, it is the first to adopt a diffusion-based generative framework in 4D face modeling. (2) We train a DDPM unconditionally and develop several downstream tasks by conditioning the reverse process. In addition to improving the efficiency of training, this paradigm makes the approach highly versatile and easily applicable to other downstream tasks. (3) In various evaluations, the landmark sequence generation and landmark-guided mesh deformation outperform SOTA methods.

3.3 Related work

Deep generative models[122, 124, 69, 202, 224] have proven effective at high-quality image synthesis, such as content-preserving image rendering with different styles, and the generation of images depicting learned objects. For 2D images, these models have also shown to be beneficial to facial expression transfer and expression editing tasks. However, the majority of existing solutions address the problem of *static* expression synthesis. Here we review some recent advances achieved in *dynamic* facial expression generation, i.e. modeling and predicting the temporal evolution of poses elicited by facial expressions.

2D facial expression video generation. There is substantial literature addressing the problem of 2D facial expression video generation[22, 242, 259, 258, 174]. MoCoGAN [242] decomposes the video into content and motion: An image-based generator creates the content and GRUs generate the motion. G^3AN [258] presents a GAN-based generative model, which also disentangles the appearance and motion of facial video and generates videos by using a spatio-temporal fusion architecture. [256] generates image sequences by using landmark sequences as guidance. Such a landmark-based approach has been also adopted in [174] where a GAN is trained over dynamic facial expressions by deploying manifold-valued representations.

Dynamic 3D facial expression synthesis. To our knowledge, dynamic 3D facial expression synthesis has not been fully explored. [187] synthesizes realistic high resolution facial expressions by using a deep mesh encoder-decoder like architecture to estimate the displacements which are then added to a neutral face frame. [216] deploys LSTMs to estimate the facial landmark changes, which are then used to guide the deformation of a neutral mesh via a Radial Basis Function network. However, both works focus on the displacement estimation for a given expression and do not consider conditional generations. The closest work to ours is Motion3DGAN [175] which extends the aforementioned MotionGAN [174] to model the dynamics of 3D landmarks. The learned distribution of 3D expression dynamics by a WGAN over the hypersphere space is sampled with a condition to generate landmark sequences, which are then fed into a mesh decoder to deform a neutral 3D face mesh frame-by-frame. Our work has several advantages over their work. First, benefiting from the power of diffusion models, we model the input distribution without requiring any extra preprocessing, and can learn from sequences of different lengths. Second, our framework offers a highly versatile and efficient alternative, as we train a DDPM unconditionally and different conditional generations can be performed solely during the reverse process in a plug-and-play manner. Finally, our landmark driven mesh deformation takes into account the identity shape of the input facial mesh and adapt the per-vertex displacements to it, generating a personalized deformation for any given input face.

Given the scarcity of existing work on 3D facial animation generation, we compare our work with some generator models originally dedicated to human motion synthesis, including Action2motion [76] and ACTOR [182].

3.4 Method

At the core of our approach is a DDPM-based model to generate a 3D landmark sequence $x = \{L_1, \ldots, L_F\}$ where a frame $L_f \in \mathbb{R}^{N\times 3}$ (for f =1 to F) represents the

3D coordinates of N landmarks. Note that the 3D arrangement of a landmark set L_f implicitly encodes the geometric information specific to the facial anatomy of an individual, and can be viewed as a mixture of the facial identity shape at a neutral pose L and the pose-induced shape change, i.e. $L_f = \Delta L_f + L$. The method is composed of two tasks: First, a DDPM is trained unconditionally (Sec.3.4.1), whereas conditional generations are obtained by conditioning the reverse process. Different forms of conditioning can be performed, leading to several downstream tasks (Sec.3.4.2). Then, our landmark-guided encoder-decoder (Sec.3.4.3) estimates ΔM_f at each frame (for f =1 to F), using a target neutral face mesh M and ΔL_f as input. The desired animation mesh sequence $\{M_1, \ldots, M_F\}$ is obtained by adding the estimated displacement ΔM_f to M at each corresponding frame, i.e. $M_f = M + \Delta M_f$. The overview of the proposed method is illustrated in Fig.5.1.

Note that directly training from and generating full meshes may be beneficial but raises technical issues since the model becomes computationally and memory intensive. An alternative is to utilize diffusion models directly in the latent space of autoencoders [205], or a pre-constructed parameter space of 3D face. Our work can be viewed as akin to the latter approach, except that we use a heuristically defined feature space, i.e., the landmark space, instead of a learned latent space. This choice has been validated by the quality of the reconstruction obtained by the landmark-guided encoder-decoder (Tab. 3.5).

3.4.1 Denoising Diffusion Probabilistic Models

DDPMs are latent variable models where the latent variables x_t (for t =1 to T) have the same dimension as the original data $x_0 \sim q(x_0)$. In our work, x_0 is a landmark-based facial animation data: $x_0 = \{L_1, \ldots, L_F\}$. Note that it is contrary to most prior works which generate only the displacements ΔL_f [174, 175, 216]. Training our model to generate L_f directly allows it to learn to produce quality expressions that are consistent with the inherent facial morphology.

The joint distribution $p_\theta(x_{0:T})$ from which we derive the likelihood $p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}$ is called the reverse process whereas the approximate posterior $q(x_{1:T}|x_0)$ is called the forward process or diffusion process. The diffusion process produces gradually noisier samples $(x_1, x_2, \ldots x_T)$ by adding Gaussian noise to the initial data x_0 according to a variance schedule $\beta_1, ..., \beta_T$ [89]:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{3.1}$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I). \tag{3.2}$$

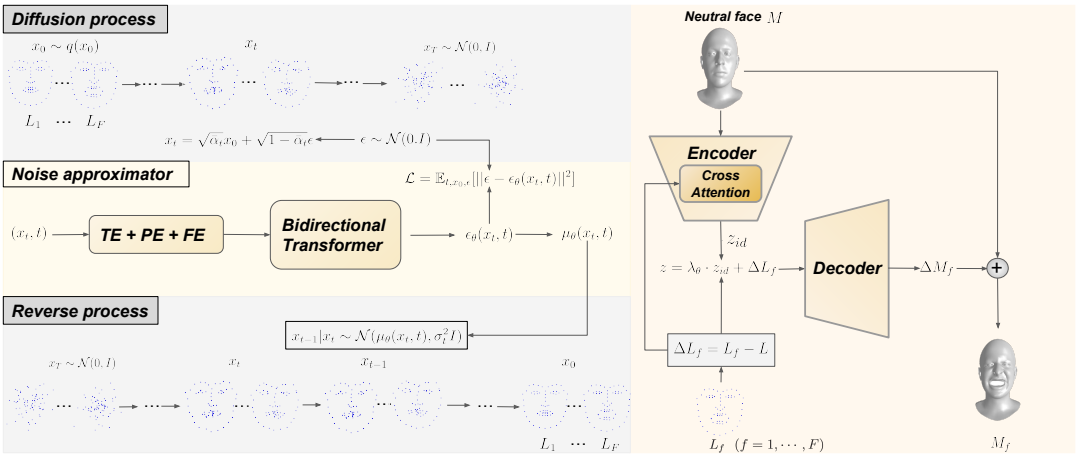


Fig. 3.1 Overview of the proposed approach. Generally, the diffusion process is used to train the noise approximator while the reverse process is used to sample x_0 from the distribution q. But some tasks developed in Sec. 3.4.2 require both processes for sampling. The bidirectional transformer takes as input the sum of the outputs of three embedding layers: the temporal embedding layer (TE) that takes as input t, the positional encoding layer (PE) that takes as input an integer sequence from 1 to F, and the feature embedding layer (FE) that takes x_t. The landmark-guided encoder-decoder retargets the expression of L_f onto the input mesh M to estimate M_f at each frame.

We can derive from Eq. 3.2 the following property [89] which allows us to train the diffusion model efficiently at an arbitrary time step t:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I), \tag{3.3}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\alpha_t = 1 - \beta_t$.

x_T follows a near-isotropic Gaussian distribution provided that a well-behaved schedule is defined and that T is sufficiently large. DDPM [89] uses this property to sample the target distribution q ($x_0 \sim q(x_0)$). This is achieved by reversing the diffusion process: It begins by sampling x_T from $\mathcal{N}(0, I)$. Next, the reverse process generates progressively less-noisy samples $x_{T-1}, x_{T-2}, \ldots, x_1$ until $x_0 \sim q(x_0)$ is obtained, by repeatedly sampling x_{t-1} from $p_\theta(x_{t-1}|x_t)$ by using Eq. 3.5. This reverse process is formally defined as a Markov chain with learned Gaussian transitions whose mean and variance are estimated by a neural network of parameter θ:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{3.4}$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{3.5}$$

where $p(x_T) = \mathcal{N}(x_T; 0, I)$. As in [89], we set $\Sigma_\theta(x_t, t)$ to $\sigma_t^2 I$. This is a reasonable

choice for generating quality samples, provided that T is chosen to be sufficiently large [170]. Note that estimating $\Sigma_\theta(x_t, t)$ allows sampling with many fewer steps [170].

Several possibilities can be considered to parameterize $\mu_\theta(x_t, t)$ in Eq. 3.5. [89] shows that approximating the noise ϵ that appears in the following equation:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \tag{3.6}$$

is a suitable choice, especially when combined with a simple loss function (See Eq. 3.8). Note that Eq. 3.6 is a different way of writing Eq. 3.3 ($\epsilon \sim \mathcal{N}(0, I)$). Finally, the term $\mu_\theta(x_t, t)$ can be computed from the approximation of ϵ, denoted as $\epsilon_\theta(x_t, t)$:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta \left(x_t, t \right) \right). \tag{3.7}$$

Diffusion models can be trained by optimizing the usual variational bound on negative log-likelihood, but we adopt here the simplified objective function proposed in [89]:

$$\mathbb{E}_{t, x_0, \epsilon}[||\epsilon - \epsilon_\theta(x_t, t)||^2], \tag{3.8}$$

where the term x_t is computed from Eq. 3.6.

Many previous works [89, 170, 205, 46], especially those for modeling 2D images, have utilized a UNet-like structure[206] to model the mean $\mu_\theta(x, t)$ or the noise $\epsilon_\theta(x, t)$. Here we employ a bidirectional transformer (BiT) [45] to efficiently capture the temporal characteristics of x_t.

3.4.2 Downstream tasks

The DDPM is learned unconditionally and several downstream tasks are developed from the same learned model, such as expression control (with label or text), expression filling (with partial sequence), or geometry-adaptive generation (with facial geometry). The pseudo code for each task can be found in Sec. 3.9.4.

Conditioning on expression label (label control). The task is to perform a conditional generation according to the expression label y. Conditioning the reverse process of an unconditional DDPM is achieved by using the classifier-guidance [231, 46, 139]. First, we train a classifier that predicts the label y given a latent variable x_t (and t). Here the classification is conducted with a BiT [45] by adopting the usual approach of adding an extra learnable classification token [45]. Note that the BiT presented here should be distinguished from the other BiT in the diffusion model and is used to condition its

reverse process. It is achieved by sampling x_t according to the distribution:

$$p_{\theta,\phi}(x_t|x_{t+1}, y) \propto p_\theta(x_t|x_{t+1})p_\phi(y|x_t), \tag{3.9}$$

where ϕ represents the parameters of the classifier. Sampling of Eq. 3.9 can be achieved approximately [224] by sampling from a Gaussian distribution similar to the unconditional transition operator $p_\theta(x_t|x_{t+1})$, but with its mean shifted by a quantity proportional to $\Sigma_\theta(x_t, t)\nabla_{x_t} p_\phi(y|x_t)$.

Instead of sampling Eq. 3.9, we used an alternative way, as proposed in [139]: x_t is computed so as to maximize the log of Eq. 3.9. A hyperparameter λ is used to adjust the trade-off between fluency ($p_\theta(x_t|x_{t+1})$) and control ($p_\phi(y|x_t)$), leading to a stochastic decoding method that balances maximizing and sampling $p_{\theta,\phi}(x_t|x_{t+1}, y)$. As in [139], optimization is achieved by running 3 steps of the Adagrad [53] update for each diffusion step (Alg. 1 of Sec. 3.9.4).

Conditioning on text (text control). We also use in this task a BiT guidance, but instead of estimating a label from x_t and t, the BiT outputs a vector of dimension 512 (the softmax layer is removed). As in [236], the BiT is trained so as to increase the cosine similarity between its output and the textual features extracted with CLIP [191] from the text associated with x_0.

Conditioning the reverse process according to the text c is then achieved (Alg. 2 of Sec. 3.9.4) by adapting the procedure presented for the label control: x_t is computed so that it maximizes:

$$\lambda \cdot log(p_\theta(x_t|x_{t+1})) + cos(\text{BiT}(x_t, t), \text{CLIP}(c)). \tag{3.10}$$

Conditioning on partial sequence (expression filling). Similarly to inpainting whose purpose is to predict missing pixels of an image using a mask region as a condition, this task aims to predict missing frames of a temporal sequence by leveraging known frames as a condition. The sequence x_0 is composed of F frames, which are either known or unknown. Let $\mathcal{S}_K$ and $\mathcal{S}_U$ denote respectively the set of indices associated with known and unknown frames, and let $x|_\mathcal{S}$ denote the subsequence containing only the frames of x whose indices belong to $\mathcal{S}$.

Since $x_0|_{\mathcal{S}_K}$ is known, note that $x_t|_{\mathcal{S}_K}$ can be drawn according to Eq. 3.3. Indeed, each component of x_t can be drawn independently since $q(x_t|x_0)$ is an isotropic normal distribution. Sampling from the reverse process conditioned on a partial sequence can also be achieved as follows: X_T is first determined: $x_T|_{\mathcal{S}_U}$ is drawn from $\mathcal{N}(0, I)$ and $x_T|_{\mathcal{S}_K}$ according to Eq. 3.3. Then, computing x_t from x_{t+1} is achieved in two steps: First, a temporal sequence $\hat{x}_t$ is simply drawn from $p_\theta(.|x_{t+1})$ (it is the way to compute x_t in the usual case). $x_t|_{\mathcal{S}_U}$ is set to $\hat{x}_t|_{\mathcal{S}_U}$, while for known frames, $x_t|_{\mathcal{S}_K}$ is directly

drawn according to Eq. 3.3 (Alg. 3 in Sec. 3.9.4). Despite its simplicity, this strategy gives satisfactory results as we will demonstrate through qualitative validation in later sections of this paper, provided that the partial sequence is of sufficient length.

Geometry-adaptive generation. Given the facial geometry of a specific subject, a generation can be performed as a special case of expression filling: $\mathcal{S}_K$ is set to $\{1\}$ or to $\{F\}$ (F is the sequence length) and the unique known frame associated with $x_0|_{\mathcal{S}_K}$ is set to the neutral face L of the subject. The remaining sequence is considered as unknown, for which the model performs an expression filling.

However, we observed that the generated frames may not always smoothly connect to the given frame, a problem that did not arise when the partial sequence remained long enough. In the context of image inpainting, [155] also shows that the simple sampling strategy used for the expression filling task may introduce disharmony. A more sophisticated approach has been proposed so as to harmonize the conditional data $x_t|_{\mathcal{S}_K}$ with the generated one $x_t|_{\mathcal{S}_U}$ [155]. In order to achieve better convergence properties of the algorithm while maintaining its simplicity, we derive the sequence with five iterations, each with a slight modification: For the first iteration, $x_T|_{\mathcal{S}_U}$ is drawn, as previously, from $\mathcal{N}(0, I)$. For the following iterations, $x_T|_{\mathcal{S}_U}$, as $x_T|_{\mathcal{S}_K}$, is drawn according to Eq. 3.3 where x_0 is the result obtained from the previous iteration. By doing so, we expect $x_T|_{\mathcal{S}_U}$ and $x_T|_{\mathcal{S}_K}$ to be harmonized progressively, thus leading to the improved harmonization of $x_t|_{\mathcal{S}_U}$ and $x_t|_{\mathcal{S}_K}$ along the iterations.

Note that this process can also be easily guided by a classifier (as in the label control) so as to generate a desired facial expression starting from a given facial anatomy (See Alg. 4 in Sec. 3.9.4). In this case, the method used for the expression filling must be modified as follows: In the expression filling task, a sequence $\hat{x}_t$ was drawn from $p_\theta(.|x_{t+1})$. In order to guide the reverse process, the sequence $\hat{x}_t$ can now be estimated so as to maximize $\lambda \cdot \log p_\theta(x_t|x_{t+1}) + \log p_\phi(y|x_t, t)$, similarly to the label control case.

3.4.3 Landmark-guided mesh deformation

To obtain the full mesh sequence $\{M_1, \ldots, M_F\}$ from $\{L_1, \ldots, L_F\}$, one could use existing fitting methods such as FLAME [138] or DL-3DMM [61] so as to preserve both the facial anatomy and the expression encoded in the landmark frames. However, the meshes generated through the linear blending models tend to lack intricate details of facial geometry, resulting in dull, lifeless shapes. Thus, in our work, we retarget the expression encoded in L_f to the facial geometry given as a (realistic) input mesh M, as in [175]. The mesh M is assumed to be at its neutral pose with a predefined topology [138]. Each mesh frame M_f should retain the facial identity shape M, combined with the expression-driven shape change encoded in $\Delta L_f = L_f - L$ (ΔL_f represents the landmark displacement at f-th frame). This is achieved by our encoder-decoder net-

work that takes both M and ΔL_f as input and predicts ΔM_f at each frame, which is respectively added to M to obtain the final mesh sequence: $M_f = M + \Delta M_f$. This is similar to the Sparse2Dense mesh decoder proposed in [175], except that only ΔL_f (and not M) is used to predict ΔM_f in their work. In our approach, on the other hand, we take into account the different morphological shapes of the neutral mesh M to adapt the estimation of per-vertex displacements ΔM_f.

In order to benefit from the consistent and quality expressions adapted to the facial morphology by the DDPM, one can extract a landmark set L_M from a mesh M, perform the geometry-adaptive task on it to generate a sequence involving L_M, and retarget it to M by the landmark-guided mesh deformation.

Encoder and decoder. Inspired by the Sparse2Dense mesh decoder of [175], we develop an encoder-decoder architecture based on spiral operation layers. The encoder contains a backbone consisting of five spiral operation layers [20] that extracts the features of M. In addition, we propose to incorporate a cross-attention mechanism [248] to account for the possible influence of the characteristics of M on the impact of ΔL_f on each vertex of M: It enables us to find the relevant features of the mesh M that can help predict a latent representation (of ΔM_f) according to ΔL_f. More specifically, the *query* is derived from a linear embedding of ΔL_f (computed by a fully-connected layer FC) and the *key, value* pairs from the output of the backbone (i.e. features of M) denoted as F. The output of the attention layer writes:

$$softmax\left(\frac{FC(\Delta L_f)\cdot F^T}{\sqrt{d}}\right)F, \tag{3.11}$$

where d is the dimension of F. Then a linear layer maps the vector of Eq. 3.11 to the *identity-aware* representation z_{id}, which is further shifted by the landmark displacement ΔL_f to obtain the final latent representation: $z = \lambda_\theta \cdot z_{id} + \Delta L_f$, where the weight parameter λ_θ is a learnable parameter.

The decoder consists of a linear layer and five spiral operation layers. It takes the latent representation z as input and outputs the per-vertex displacement ΔM_f. M_f is then set to $M + \Delta M_f$. The model is learned using the loss function proposed in [175].

3.5 Experimental setting

As proposed in [89], we set a linear noise schedule starting from $\beta_1 = 1e-4$ to $\beta_T = 0.02$, and σ_t^2 is set to β_t. T is set to 2000. We train the model on 200K iterations with a learning rate of $1e-4$ and a batch size of 256. The hyperparameter λ that is used to guide the sampling of the reverse process is set to 0.01 as in [139].

CoMA dataset [197] is a commonly used 4D facial expression dataset in face modeling tasks [21, 108], consisting of over a hundred 3D facial animation sequences cap-

tured from 12 subjects, each performing 12 facial actions ("high smile", "mouth up", etc.). Each data is composed of a triangular mesh of 5023 vertices undergoing some deformation elicited by an expression.

BU-4DFE dataset [288] contains a total of 606 sequences of 83 landmarks extracted from a sequence of 3D facial scans. Six basic emotional expressions ("anger", "disgust", "fear", "happy", "sad", and "surprise") of 101 subjects have been recorded.

Different sequences have been used depending on the specific task at hand. Unless otherwise specified, solely the sequences from the CoMA dataset have been utilized.

3.6 Results involving various conditional generations

Here we describe the results we obtained on the various conditional generations. Throughout this section, a classifier that predicts the expression from a sequence independently of its type (see Section 3.5) is called a classifier of type I (order-**I**nsensitive), whereas a classifier of type S (order-**S**ensitive) predicts both the expression class and the expression type (either N2E or E2N).

For evaluation purposes, an independent classifier (which we denote as IC) is trained to predict the label from a sequence x_0. We use one LSTM layer followed by a linear layer, as in [175]. The model's ability to generate a desired expression is assessed by the classification accuracy of the IC tested on the generated expressions. Additionally, the quality of the generated sequence is assessed by using the Frechet Inception Distance (FID) score [85], that compares the distribution of fake data with that of real data. It is computed from the output of the linear layer of the IC.

3.6.1 Label control

The proposed approach is compared with several SOTA methods which perform conditional sequence generation: Action2Motion [76], Motion3DGAN [175] and ACTOR [182]. The BiT-based classifier used to guide the reverse process, as well as the IC are of type I. Quantitative results as measured by the classification accuracy and the FID score are summarized in Table 3.1, which confirms that the proposed approach outperforms all SOTA methods. Fig. 3.2 shows some illustrative results: Our model generates various realistic and quality expressions adapted to various facial geometries. Videos presented in the project website (`https://github.com/ZOUKaifeng/4DFM`) demonstrate the generated expressions, as well as qualitative comparisons among these methods: Sequences generated by our approach are more expressive. The diversity of the generated sequences in terms of both expression and facial anatomy is also illustrated in Sec. 3.9.3.

Table 3.1 Performance of different methods for generating desired expressions has been evaluated by measuring the classification accuracy and the FID score. We report as ground truth the FID and the accuracy computed on the test dataset, assuming that an ideal method could have generated it.

	CoMA		BU-4DFE	
Model	Acc	FID	Acc	FID
Ground truth	83.78%	2.77	99.51%	6.02
A2M	52.36%	29.44	80.83%	19.64
MoGAN	80.76%	7.72	99.26%	13.29
ACTOR	81.40%	7.11	99.13%	14.56
Ours	84.97%	6.79	99.89%	12.37

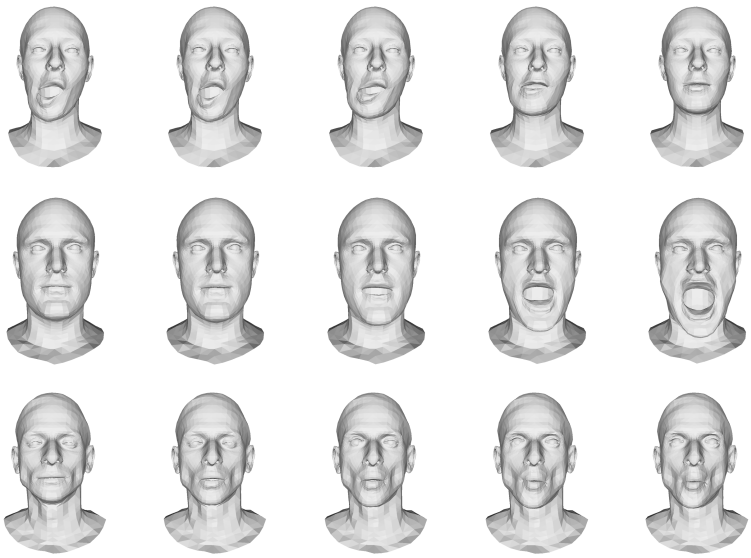


Fig. 3.2 Animated mesh sequences guided by the label "mouth side" (top), "mouth extreme" (middle), and "cheeks in" (bottom). The meshes are obtained by retargeting the expression of the generated x_0 on different neutral faces.

3.6.2 Text control

To demonstrate this task, we have increased the vocabulary of our dataset by merging CoMA and BU-4DFE. In the first experiment, the raw text label is used to condition the animation (we call it *raw text* task) and the IC used for the evaluation is of type I. In the second experiment, the description of a sequence is enriched to be a short sentence such as "from the neutral face to the raw text label", or "from the raw text label to the neutral face" (we call it *enriched text* task) and the IC used for the evaluation is of type S.

We compare our results with those of MotionClip [236]. Quantitative results are shown in Table 3.2. Classification accuracies obtained with the proposed method are

slightly higher than those of MotionClip, with FID scores significantly lower. Sequences created by MotionClip are actually realistic but the FID scores are high, due to the lack of diversity in the generated sequences.

Fig. 3.3 shows illustrative examples obtained with the proposed approach. Note that our model is able to create animated meshes that combine different types of expressions by compositing a text combining different types of expressions. For the complete sequences as well as the qualitative comparisons, readers may refer to the project website.

Table 3.2 Quantitative evaluation of the text control task. Classification accuracy and FID are computed for the raw text task (rtt, left) and for the enriched text task (ent, right).

	Acc (rtt)	FID	Acc (ent)	FID
Ground truth	86.02%	3.67	74.40%	4.56
MotionClip	80.67%	42.19	58.33%	38.83
Ours	82.01%	9.46	64.38%	11.34

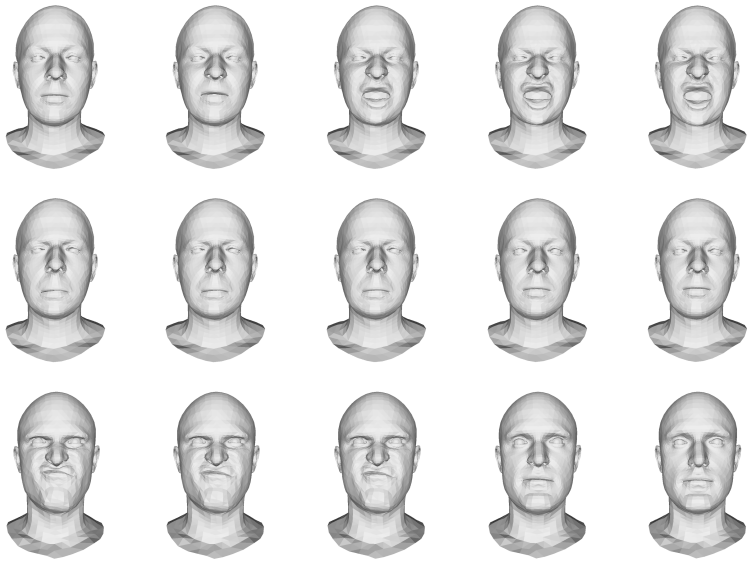


Fig. 3.3 Text-driven generation results obtained by the *enriched text* task ("from neutral face to bareteeth" (top)), and by the *raw text* task ("angry mouth down" (middle), "disgust high smile" (bottom)). The input texts used for the *raw text* task are the combinations of two terms used for training. For instance, "disgust high smile" is a new description that hasn't been seen before, which combines "disgust" and "high smile".

3.6.3 Expression filling

Given a partial sequence of an expression, the model can fill up the missing frames. Three experiments have been conducted: In the filling from the beginning (FFB) or the filling from the end (FFE) cases, the length l of the partial sequence is drawn uniformly

in $[10, 30]$. In the filling from the middle (FFM) case, l frames have been given at the beginning and at the end of the sequence, respectively. l is uniformly sampled in $[5, 15]$.

The proposed approach for expression sequence filling is compared with a mean imputation strategy. To evaluate the result, an IC (of type I) is trained, so as to check if the filled data has the same expression class as the original one. Results are shown in Table 3.3. The expression label of the partial sequence is well-captured and reflected in the filled part, leading to an improved classification accuracy especially for the FFM case, where the classification accuracy is comparable to that obtained for the ground truth (Table 3.1). Classification accuracies obtained in the FFE and FFB cases are lower due to the content of the sequences. As an example, when the partial sequence is associated with the beginning of a sequence of type N2E, it may be composed, at worst, of neutral faces only, or at best of less expressive faces. This is worsened by the fact that sometimes certain expressions appear only at the end of the sequences. This is contrary to the FFM case, where the partial sequence contains both the neutral and the most expressive poses.

Finally, there is a significant improvement of FID score after filling with the proposed approach. Furthermore, our videos presented on the project website illustrate that the generated sequences are smoothly connected to the given partial sequence.

Table 3.3 Quantitative evaluation of the expression filling task for three different locations of the missing part. Accuracy and FID are computed on the sequences obtained by the mean imputation strategy, and by our diffusion model. Note that accuracy is 83.78% and FID is 2.77 for the ground truth in all cases (FFE, FFM, FFB).

	Mean Imputation		Ours	
	Acc	FID	Acc	FID
FFE	60.15%	25.67	67.18%	5.51
FFM	56.25%	17.68	85.93%	5.06
FFB	53.90%	27.32	70.31%	5.22

3.6.4 Geometry-adaptive generation

We have conducted the geometry-adaptive generation task by using classifier guidance so as to generate a desired facial expression from a given facial anatomy (Alg. 4 of Sec. 3.9.4). The BiT used for guidance and the IC used for evaluation are both of type S. $\mathcal{S}_K$ is set to $\{1\}$ if the chosen label is associated with N2E sequences, and to $\{F\}$ otherwise.

Quantitative results are shown in Table 3.4. The classification accuracy is close to the ground truth, and the visual inspection of the video sequences on the project website shows no gap between the generated frames and the enforced one.

Table 3.4 Quantitative evaluation of the geometry-adaptive generation task.

	Acc	FID
Ground truth	71.01%	5.57
Geometry-adaptive	70.43%	9.26

While Sec. 3.9.3 illustrates the diversity of generated expressions when the model is conditioned on the expression label, we study here the same type of diversity but when the facial geometry of a specific subject is enforced in the conditioning process. To this end, a landmark set L_M has been extracted from a given mesh M. The geometry-adaptive generation task is performed so as to generate a sequence containing L_M, and exhibiting an expression corresponding to a given label y. Then, the generated sequence is retargeted to M with the landmark-guided mesh deformation.

Fig. 3.4 illustrates the variety of expressions we thus obtained by using a same facial anatomy L_M and a same label y (either "eyebrow" or "high smile"), which confirms that the proposed approach is able to generate expression sequences with sufficient level of diversity, even if a same facial anatomy is used for conditioning.

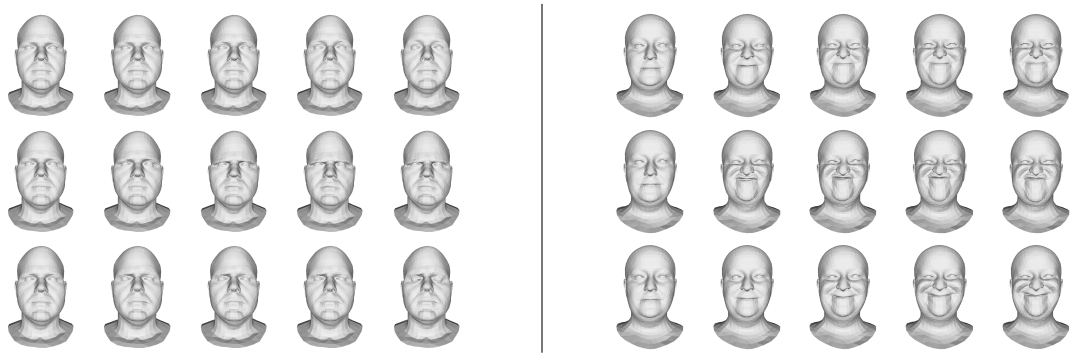


Fig. 3.4 Diversity of expressions generated with the label "eyebrow" (left), and "high smile" (right) in the geometry-adaptive generation task. All illustrated sequences are of type N2E. Note that eyebrows can be either lowered (the second and third rows) or raised (the first row). Although the poses of maximal expression intensity look all similar in the three sequences of "high smile", their temporal properties are significantly different.

3.7 Results related to landmark-guided mesh deformation

3.7.1 Comparison with other methods

To the best of our knowledge, only [175] and our work estimate M_f from M and ΔL_f. Note that both approaches use spiral convolution. For the comparative experiments, we also adapt two autoencoders: CoMA [197], which uses Chebyshev convolution and a mesh pooling, and the autoencoder proposed in [29] (the encoder and decoder

consisting of three layers of linear, nonlinear, and linear activation units, respectively). Both decoders, which originally take the latent representation of the input mesh as input, have been modified so as to consume the concatenation of the latent representation with ΔL_f.

We conducted two series of experiments: Either 3 expressions (expression split) or 3 subjects (identity split) have been excluded from the training set, and the performance of the model is evaluated on the excluded data. The mean per-vertex Euclidean error between the generated meshes and their ground truth has been measured to assess the performance. Quantitative results are shown in Table 3.5. While the three methods based on spiral convolution generally yield effective results, our approach outperforms the others, thus confirming the advantage of the cross-attention layer, in particular.

Table 3.5 Per-vertex reconstruction error (mm).

Method	Expression split	Identity split
Linear [29]	0.67 ± 0.76	0.73 ± 0.77
CoMA[197]	0.58 ± 0.63	0.63 ± 0.67
S2D [175]	0.52 ± 0.59	0.55 ± 0.62
Ours(w/o attention)	0.54 ± 0.59	0.57 ± 0.64
Ours	0.45 ± 0.51	0.50 ± 0.58

We propose to complement our quantitative analysis by a qualitative comparison of the different methods. As the "Expression split" and "Identity split" experiments yield very similar results, we focus solely on the "Identity split" experiment in the following.

Fig. 3.5 depicts the ground truth mesh (a) as well as the meshes generated with several approaches (b-e). Each vertex of a generated mesh is assigned a color representing the Euclidean distance to its counterpart on the ground truth mesh. As expected, the errors appear mainly on the regions that have been deformed to attain the expression. In Fig. 3.5, retargeting an expression close to the neutral pose (first row) leads to tiny errors, whereas retargeting an expression "mouth extreme" leads to errors that are mostly located near the mouth. Our approach achieves the best performance in this qualitative error measure, confirming the quantitative results described above.

3.7.2 Expression retargeting

Our landmark-guided encoder-decoder can retarget the landmark expression sequences to different facial meshes. In Fig. 3.6, a landmark sequence generated from our model is used to guide the deformation of three different facial meshes. We can observe that different subjects make the same semantic facial expression in response to the same chosen landmark sequence. As expected, the resulting mesh deformations are well adapted to each facial geometry, which confirms that our model offers the flexibility of combining any desired facial meshes independently from the landmark sequence

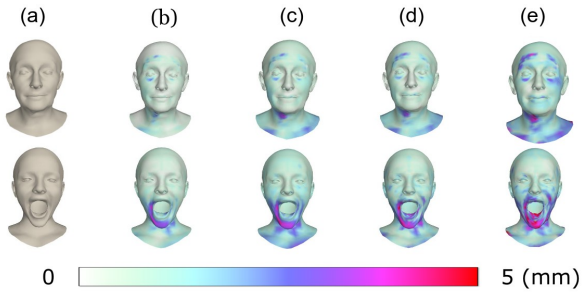


Fig. 3.5 Qualitative comparison of our method (b) with S2D (c), CoMA (d), and Linear (e) in the landmark-guided deformation of a given mesh. The ground truth meshes are given in the first column (a). The expression of the first row is close to the neutral face and that of the second row is taken from a sequence labeled as "mouth extreme".

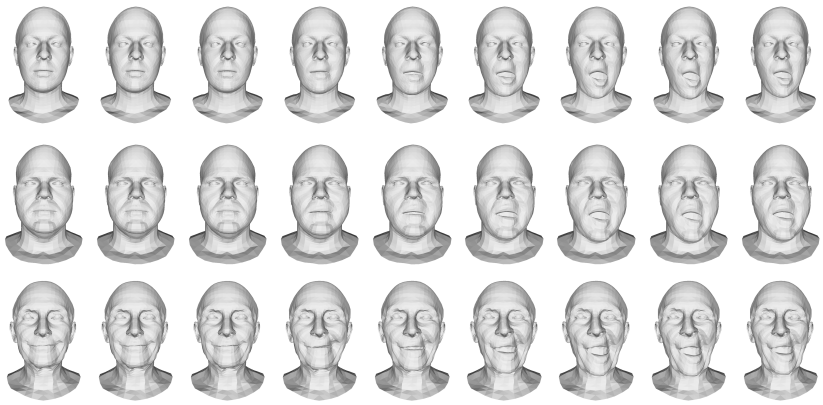


Fig. 3.6 Expression retargeting results by our landmark-guided encoder-decoder. A same expression sequence (generated by using "mouth side" label) has been applied to three different facial meshes.

generation. More results can be found on our project website, where we illustrate the retargeting results of the landmark sequence taken from a full sequence of the CoMA dataset onto several facial meshes.

3.8 Conclusion

We have presented a generator model to synthesize 3D dynamic facial expressions. The dynamics of facial expressions is first learned unconditionally, from which a series of downstream tasks are developed to synthesize an expression sequence conditioned on various condition signals. Also proposed is a robust face deformation scheme guided by the landmark set, which contributes to a higher reconstruction validity. Experimental results show that the proposed method can produce plausible face meshes of diverse types of expressions on different subjects. In addition, it outperforms SOTA models both qualitatively and quantitatively. As has been demonstrated, our expression generation framework is versatile and can be used in many application scenarios including,

but not limited to, label-guided generation, text-driven generation, geometry-adaptive generation, or expression filling.

3.9 Annexes

3.9.1 Advantage of using a bidirectional Transformer

In order to efficiently capture the temporal features of x_t, we use a bidirectional Transformer (BiT) as the noise approximator as well as the classifier used for the guidance. We compare the performance of the bidirectional Transformer to other popular neural networks such as Transformer [248] and U-Net [206], the most frequently used model for 2D images.

We use a 1D U-Net that takes as input a tensor of size *channels*=40 and *num_features*=68×3, where 40 is the sequence length, and 68 the number of 3D landmarks. These models are evaluated in the context of the label control task on the CoMA dataset, as detailed in Sec. 3.6.1. Results are given in Tab. 3.6.

Table 3.6 Quantitative evaluation of the label control task. The noise approximator and the classifier used for the guidance are modeled either with a U-Net, a Transformer, or a BiT.

Model	Accuracy	FID
U-Net	50.04%	21.36
Transformer	80.29%	7.57
BiT	84.97%	6.79

As expected, U-Net is not adapted to temporal sequence modeling. We observe that the best results are obtained by using a BiT.

3.9.2 Training with sequences of any length and generation of sequences of arbitrary length

Since our noise approximator is a bidirectional transformer, it can take sequences of any length as input —It can be trained using sequences of any length, and we can sample from the resulting model so as to obtain sequences of desired lengths (The length of x_0 will be that of x_T). In the same way, as a bidirectional transformer is used also to guide the reverse process, it can guide the reverse process with any length for x_t. Consequently, tasks related to label control, text control, and geometry-adaptive generation can generate sequences of any desired length. Furthermore, the sequences that have to be filled with the expression filling task can be of any length.

For the sake of simplicity, we describe here only the label control task. The noise approximator and the classifier used for the guidance are either trained using sequences

of a fixed length ($F = 40$) or variable lengths (F is uniformly distributed in the interval $[35, 45]$).

The performance of both models is evaluated when outputting sequences of length in $[35, 45]$. The performance is evaluated as in Sec. 3.6.1, except that the independent classifier is trained with sequences of variable length (F is uniformly distributed in $[35, 45]$). Results are shown in Fig.3.7.

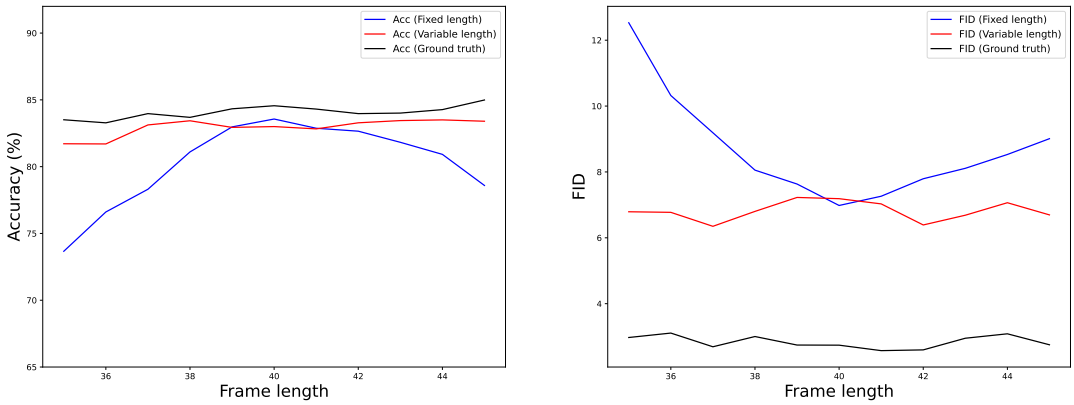


Fig. 3.7 Quantitative evaluation of the label control task for models trained with sequences of a fixed length ($F = 40$) or variable lengths. Performance is evaluated on generated sequences of different lengths using, as in Sec. 3.6.1, the classification accuracy (left) and the FID score (right).

When generating sequences of different lengths is required, training with variable lengths helps the model to perform better. Moreover, the results obtained with the model trained with sequences of variable length are satisfactory: the achieved accuracy is similar to that of the ground truth. Moreover, the FID obtained for a length frame of 40 is similar to that calculated with the model dedicated to output sequences of length 40.

3.9.3 Diversity of the generated sequences when conditioning on expression label

We study in this section the diversity of the generated sequences both in terms of facial anatomy (L) and in terms of expression (ΔL_f) in the label control task. As a reminder, the 3D arrangement of a landmark frame L_f can be regarded as the combination of the facial anatomy (at a neutral pose L) and the expression-driven shape change applied to it, i.e. $L_f = \Delta L_f + L$.

Since the proposed landmark-guided mesh deformation retargets the expression $\Delta L_f = L_f - L$ onto a new face anatomy given as a mesh M, it is used hereafter to illustrate the diversity of the generated expressions but it is not adapted to analyze the facial anatomy of the generated L. To show the diversity of facial anatomy gener-

ated by our model, we use the FLAME model[138] to compute the facial mesh from the landmark set of neutral pose[1].

Fig. 3.8 presents three illustrative neutral faces L that we generated by conditioning the reverse process on the same expression label "mouth open". Both landmark set and the FLAME-fitted mesh are shown, for each face. (The neutral face L associated with a generated sequence x_0 is set to either L_1 or L_F, depending on the sequence type.) Additionally, the diversity in the generated expression is illustrated in Fig. 3.9. The apparent distinction among these results demonstrate that the proposed approach is able to generate sequences of rich diversity, both in terms of facial anatomy and expression (This is due to the input noise x_T that is sampled from $\mathcal{N}(0, I)$).

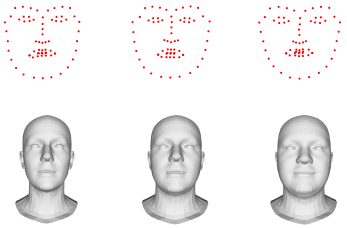


Fig. 3.8 Diversity of facial anatomy in the generated expressions. We use FLAME model to compute facial meshes from the landmark sets, for the visualization purpose.

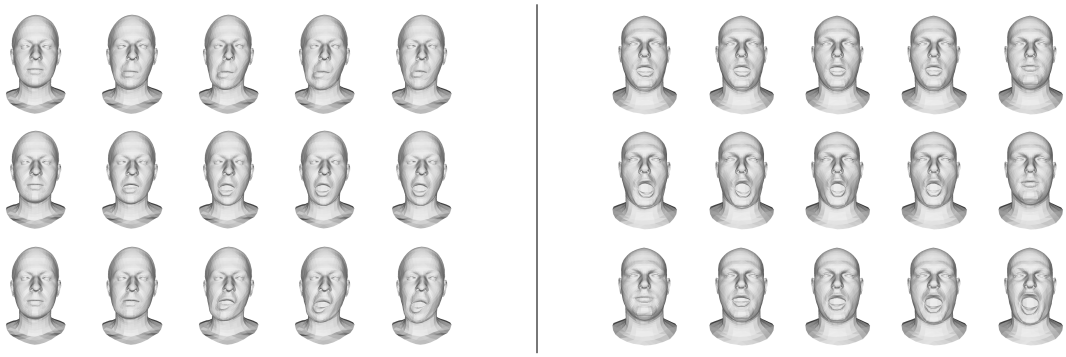


Fig. 3.9 Diversity of expressions generated with the label "mouth side" (left), and "mouth open" (right) in the label control task. Note that generated sequences can be either of type E2N or N2E.

[1]We can note that the meshes generated from FLAME lack certain details of the facial geometry, resulting in dull, lifeless shapes. Furthermore, FLAME takes about 470s to fit one sequence, while the proposed landmark-guided mesh deformation needs only about 1.30s.

3.9.4 Pseudo code for each downstream task

Algorithm 1 Label control

Input: Label y.

Output: Sequence x_0 (corresponding to label y).

$x_T \sim N(0, I)$
for $t = T, ..., 1$ **do**
▷ Estimation of $p_\theta(.|x_t)$
    Compute $\epsilon_\theta\left(x_t, t\right)$
    Compute $\mu_\theta(x_t, t)$: $\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(x_t, t\right)\right)$
▷ Sampling from $p_\theta(.|x_t)$
    $z \sim N(0, I)$ if $t > 1$, 0 otherwise
    Set $\hat{x}_{t-1}$ to $\mu_\theta(x_t, t) + \sigma_t z$
▷ Optimization: optimization procedure is initialized with $\hat{x}_{t-1}$
    $x_{t-1} = \underset{x}{\operatorname{argmax}}\left[\lambda log(p_\theta(x|x_t)) + log(p_\phi(y|x, t-1))\right]$
return x_0

Algorithm 2 Text control

Input: Text c.

Output: Sequence x_0 (corresponding to text c).

$x_T \sim N(0, I)$
for $t = T, ..., 1$ **do**
▷ Estimation of $p_\theta(.|x_t)$
    Compute $\epsilon_\theta\left(x_t, t\right)$
    Compute $\mu_\theta(x_t, t)$: $\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(x_t, t\right)\right)$
▷ Sampling from $p_\theta(.|x_t)$
    $z \sim N(0, I)$ if $t > 1$, 0 otherwise
    Set $\hat{x}_{t-1}$ to $\mu_\theta(x_t, t) + \sigma_t z$
▷ Optimization: optimization procedure is initialized with $\hat{x}_{t-1}$
    $x_{t-1} = \underset{x}{\operatorname{argmax}}\left[\lambda log(p_\theta(x|x_t)) + cos(BiT(x, t-1), CLIP(c))\right]$
return x_0

Algorithm 3 Sequence filling

Input: Partial sequence $x_0|_{\mathcal{S}_K}$

Output: Completed sequence x_0

$x_T|_{\mathcal{S}_U} \sim N(0, I)$

$x_T|_{\mathcal{S}_K} = \sqrt{\bar{\alpha}_T} x_0|_{S_k} + \sqrt{1 - \bar{\alpha}_T}\epsilon, \epsilon \sim N(0, I)$

for $t = T, ..., 1$ **do**

▷ Estimation of $p_\theta(.|x_t)$

Compute $\epsilon_\theta(x_t, t)$

Compute $\mu_\theta(x_t, t)$: $\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$

▷ Sampling from $p_\theta(.|x_t)$

$z \sim N(0, I)$ if $t > 1$, 0 otherwise

Set $\hat{x}_{t-1}$ to $\mu_\theta(x_t, t) + \sigma_t z$

▷ Computation of x_{t-1}

$x_{t-1}|_{\mathcal{S}_U}$ = $\hat{x}_{t-1}|_{\mathcal{S}_U}$

if $t > 1$ **then** ▷ if $t = 1$, $x_0|_{\mathcal{S}_K}$ is already properly set.

$x_{t-1}|_{S_K} = \sqrt{\bar{\alpha}_{t-1}} x_0|_{S_k} + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon, \epsilon \sim N(0, I)$

return x_0

Algorithm 4 Geometry-adaptive generation with label control

Input: Label y and partial sequence $x_0|_{\mathcal{S}_K}$. $\mathcal{S}_K$ is either $\{1\}$ or $\{F\}$ and the unique frame associated with $x_0|_{\mathcal{S}_K}$ is a neutral one.

Output: Completed sequence x_0 (corresponding to label y)

for $i = 1$ `to` 5 **do**
  if i == 1 **then**
    $x_T|_{\mathcal{S}_U} \sim N(0, I)$
    $x_T|_{\mathcal{S}_K} = \sqrt{\bar{\alpha}_T} x_0|_{S_k} + \sqrt{1 - \bar{\alpha}_T}\epsilon, \epsilon \sim N(0, I)$
  else
    $x_T = \sqrt{\bar{\alpha}_T} x_0 + \sqrt{1 - \bar{\alpha}_T}\epsilon, \epsilon \sim N(0, I)$
  for $t = T, ..., 1$ **do**
    ▷ Estimation of $p_\theta(.|x_t)$
    Compute $\epsilon_\theta(x_t, t)$
    Compute $\mu_\theta(x_t, t)$: $\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$
    ▷ Sampling from $p_\theta(.|x_t)$
    $z \sim N(0, I)$ if $t > 1$, 0 otherwise
    Set $\hat{x}_{t-1}$ to $\mu_\theta(x_t, t) + \sigma_t z$
    ▷ Optimization: optimization procedure is initialized with $\hat{x}_{t-1}$
    $\hat{x}_{t-1} = \underset{x}{\operatorname{argmax}} \left[\lambda log(p_\theta(x|x_t)) + log(p_\phi(y|x, t-1))\right]$
    ▷ Computation of x_{t-1}
    $x_{t-1}|_{\mathcal{S}_U} = \hat{x}_{t-1}|_{\mathcal{S}_U}$
    if $t > 1$ **then** ▷ if $t = 1$, $x_0|_{\mathcal{S}_K}$ is already properly set.
      $x_{t-1}|_{S_K} = \sqrt{\bar{\alpha}_{t-1}} x_0|_{S_k} + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon, \epsilon \sim N(0, I)$
return x_0

CHAPTER 4

Disentangled Representations: Towards Interpretation of Sex Determination from Hip Bone

In this chapter, we delve into the domain of supervised disentangled representation learning, focusing specifically on the task of sex determination from hipbone images represented as meshes. This chapter aims to achieve two goals. Firstly, it focuses on disentangling the sex-related information from other identity-related information within the latent representation. Secondly, it aims to develop a data-driven algorithm that surpasses traditional manual landmark positioning methods [55, 165, 26, 172] in automatically sex deternimation from hipbones.

Moreover, we recognize the need for interpretation in the classification results, particularly for individuals without specialized medical knowledge. To address this challenge, we propose a novel approach that diverges from conventional techniques such as Saliency maps [220], which are ill-suited for mesh classification. Instead, our method involves reconstructing the original mesh and transforming it to represent the opposite sex. By comparing these two reconstructions and highlighting the disparities, we can effectively showcase the regions of interest in the classification decision process, aiding in interpretation.

When it comes to data generation, as discussed in Chapter 2, there are multiple options to consider. Firstly, in the case of reconstructing the hip bone for a specific subject, it is essential to have encoding capabilities that can capture the identity information. Therefore, diffusion models may not be suitable for this particular task. Although it is possible to use an inverted DDIM to obtain the initial noise from a given image, there may be distortions present in the results [84].

As an alternative, we explore the use of GANs to generate the hip bone. Registering medical images is a challenging process, and GANs provide accurate reconstruction. As a result, GANs may also preserve errors from the registration process (high-frequency information), which can negatively impact the experimental results. This is particularly problematic when interpreting the region of interest for different labels.

Finally, we decide to apply VAE on this task. The generation process of VAE naturally removes high-frequency error from the registration process, while retaining the

essential and crucial information. Furthermore, the inherent nature of a VAE can also be leveraged for out-of-distribution detection, as it can effectively identify samples that do not conform to the learned distribution. Therefore, prior to conducting the experiment, we also utilize the VAE to eliminate samples that exhibit significant errors. Additionally, VAE provides us with a disentangled representation, which is highly beneficial for our analysis. This disentanglement allows us to explore the effects of specific factors of interest by generating new data with modified labels associated with those factors. Furthermore, supervised learning of VAE approaches inherently involve the use of a classifier [122], which aligns well with the requirements of our task.

Through our comprehensive methodology and analyses, we contribute to the field of disentangled representation learning for sex determination from hip bone images. Our approach not only improves classification accuracy but also provides interpretability, shedding light on the reason behind decision-making.

Following this, you will find an article titled "Disentangled representations: towards the interpretation of sex determination from the hip bone" that has been accepted by The Visual Computer journal. The authors of the article are Kaifeng Zou, Sylvain Faisan, Fabrice Heitz, Marie Epain, Pierre Croisille, Laurent Fanton, and Sébastien Valette.

4.1 Abstract

Neural network-based classification methods are often criticized for their lack of interpretability and explainability. By highlighting the regions of the input image that contribute the most to the decision, saliency maps have become a popular method to make neural networks interpretable. In medical imaging, they are particularly well-suited for explaining neural networks in the context of abnormality localization. Nevertheless, they seem less suitable for classification problems in which the features that allow distinguishing classes are spatially correlated and scattered. We propose here a novel paradigm based on Disentangled Variational Auto-Encoders. Instead of seeking to understand what the neural network has learned or how prediction is done, we seek to reveal class differences. This is achieved by transforming the sample from a given class into the "same" sample but belonging to another class, thus paving the way to easier interpretation of class differences. Our experiments in the context of automatic sex determination from hip bones show that the obtained results are consistent with expert knowledge. Moreover, the proposed approach enables us to confirm or question the choice of the classifier, or eventually to doubt it.

4.2 Introduction

In forensic medicine and anthropology, sex determination is generally carried out by manually assessing hip bone features [126]. Automatic classification algorithms are mainly guided by the knowledge of anthropologists, taking into account distances or angles measured from a few anatomical landmarks [55, 165, 26, 172]. Currently there exists a crucial need for practitioners in forensic science to understand classification results and such approaches have the advantage of providing easily interpretable results. But they are specifically tailored for hip bones, and are not well suited to sex determination from other bones or bone fragments, which may be necessary in forensic science.

We propose here an (automatic) deep learning-based classification approach that is completely data-driven, is free of expert knowledge, and is suited to sex determination from other bones or bone fragments. Regardless of these advantages, the proposed method will not be used by practitioners if they cannot interpret the classification results. However, meeting the need for understanding and explainability is far from easy with deep learning classification methods.

Neural networks-based classification methods are often criticized for their lack of interpretability and explainability. Even if there is not a clear consensus on the definition of interpretability and explainability, most methods dealing with interpretability and explainability aim to understand what the neural network has learned or how prediction is done. One common method to interpret the predictions of neural networks is to compute saliency maps (SMs) [220]. However, in the context of this application, the information extracted with SMs was difficult to interpret (examples of SMs are presented in Fig. 4.7).

To overcome this limitation, we consider here a different paradigm, based on disentangled generative representations. The main novelty of this paper is to show that disentanglement may bring a better understanding of classification results, highlighting the differences between the possible classes.

Disentangled representations allow us to reveal the effects of the factors of interest through the generation of new data obtained by changing the labels related to these factors [270]. As an example, [146] samples the latent space so as to provide insights from brain structure representations. Another model proposed in [291] can simulate brain images at different ages, providing an alternative way of interpreting the aging pattern.

We introduce a disentangled Variational Auto-Encoder (DVAE) to obtain a hip bone mesh representation, in which the sex label is disentangled from the other latent variables. In addition to providing the class of a given sample to analyze, a DVAE can also provide a reconstruction for each class, which provides supplementary informa-

tion to the user. As an example, if the input mesh is a male one, its reconstruction as a man should be similar to the input mesh and its reconstruction as a woman, on the other hand, should display interpretable differences in sex-specific regions. Moreover, by comparing the two reconstructions with the original mesh for several subjects, the user can get an insight into the morphological differences between male and female hip bones.

Although SMs and the proposed approaches provide understanding and explainability, they do not act at the same level. The SMs facilitate understanding of the decision process (related to a classification method): the purpose is to understand what the neural network has learned or how prediction is performed. An SM therefore reveals information about the classifier itself and not about the classification task. On the contrary, the proposed approach makes it possible to highlight the differences between the classes and thus provides information on the classification problem to be solved.

Finally, in addition to showing that disentanglement can bring a better understanding of classification results, we also show in this paper that feeding a binary classifier with the reconstructions provided by DVAE allows to obtain a classification method that is robust to missing data and therefore well-suited to bone fragments, which is a major advantage (compared to other existing methods) for applications in forensic medicine and anthropology.

Note that the classification approach as such is not the main contribution of this article. Indeed, sex determination from the hip bone may not be considered as challenging in terms of the classification task: the hip bone exhibits significant sexual dimorphism (note that the classification accuracy is very high (Tab. 4.2)). There are indeed strong anatomical differences between the male and female hip bones, such as the subpubic angle and the shapes of the obturator foramen, of the greater sciatic notch, of the pelvic inlet and of the symphysis.

The main contribution is the proposition that disentanglement can contribute to a better understanding of classification results. In particular, the proposed method allows the users to form their own opinions. As an example, we will see in Sec. 4.7 that the reconstructions provided by the proposed approach can sometimes allow us to confirm the choice made by the classifier, or it can also allow us to doubt its choice or even question it.

The remainder of this paper is organized as follows: after the presentation of the related works (Sec. 4.3), we briefly explain in Sec. 4.4 how hip bone meshes are obtained from CT scans. Sec. 4.5 presents the DVAE. Sec. 4.6 describes the experiments and the results and Sec. 4.7 proposes a discussion. Since the two reconstructions provided by DVAE enable the users to form their own opinions, Sec. 4.8 shows that the two reconstructions may also be useful to improve the accuracy of an independent classifier.

This section also addresses the case of missing data. In Sec. 4.9, we illustrate SMs for the proposed networks for comparison. Finally, Sec. 4.10 concludes the paper.

4.3 Related works

Interpretability and explainability of deep neural networks may be achieved in two ways.

The first paradigm, known as activation maximization or feature visualization via optimization, consists of producing intuitive visualizations that reveal the meaning of hidden layers. This is mainly achieved by finding a representative input that can maximize the activation of a layer [57, 168].

The second paradigm, known as attribution methods, looks for the network inputs with the highest impact on the network response. In the case of image models, this leads to the estimation of SM, which highlights the regions of the input image that contribute the most to the decision. Many attribution techniques are based on backpropagation. An SM is, for instance, computed in [220] by computing the derivative of the output with respect to the image. Several methods such as SmoothGrad [223] have been proposed to reduce the noise that is present in the gradient. Methods such as CAM [292] and Grad-CAM [215] combine gradients, network weights and/or activations at a specific layer. Other attribution techniques analyze how a perturbation in the input affects the output [62]. Finally, attribution techniques can also be achieved via local model approximation [203].

In medical imaging, SMs are becoming a popular approach that provides interpretability, especially when it comes to localization of abnormalities. Different sanity checks [12], such as intra-architecture repeatability, inter-architecture reproductibility, sensitivity to weight randomization [2] and localization accuracy can be used to assess the relevance of SMs. These criteria helped to justify the use of SMs in some studies such as in [12], but have also led to questions about the relevance of SMs [56, 280]. This indicates that SMs are not suited to all situations.

In our experiments, the information extracted with SMs was difficult to interpret (examples of SMs are presented in Fig. 4.7). Our hypothesis is that SMs are not easily interpretable on medical imaging classification problems in which the underlying features used by the neural network are spatially correlated, scattered and non-trivial.

Generative models are proposed here as a way of better understanding classification results. These models play a crucial role in many applications and in many common tasks of data science [289, 279, 141, 15, 173, 261, 254, 183]. Moreover, there is a key challenge to learn disentangled (generative) representations where some variables of interest (such as acquisition parameters, age, sex or pathology in medical applications) would be independently and explicitly encoded [18]. These representations can either

be obtained with Variational Auto-Encoders (VAEs) [124] or with generative adversarial networks (GANs) [69].

Probabilistic generative models, such as VAEs [124], define a joint probability distribution over the data and over latent random variables. Very few assumptions are generally made about the latent variables of deep generative models, leading to entangled representations.

Disentanglement can be achieved with VAEs in the unsupervised case [30, 146], in the (semi)-supervised case [219, 122, 291], and in the weakly-supervised case [207]. In the supervised or semi-supervised case, the factors of interest are explicitly labelled in all or in a part of the training set. In the weakly-supervised case, only implicit information about factors of interest is provided during learning.

The semi-supervised case is of primary importance because better disentangled models can be obtained under supervision [152]. In this case, the latent representation is generally divided into two parts: the non-interpretable part and the disentangled part corresponding to variables that explicitly model the factors of interest. In this context, several patterns of conditional dependency structures have been proposed [158, 219, 122].

In addition to VAE approaches, there is a substantial literature on image-to-image translation between unpaired image data using GAN [253, 147, 112, 278]. First, some methods try to map an image from one domain (e.g. smiling) to another one (e.g. neutral face). Among these methods, the best known is CycleGAN [296]. This approach is able to preserve key attributes of two different domains and allows to transform an image from one domain to another. Note that StarGAN [36] can perform image-to-image translations for multiple domains. Similar methods, inspired by dual learning, can also be used [217, 296, 277] to map the domains. Other GAN based approaches use architectures that are more similar to the VAEs [128, 181]. As an example, conditional GAN [181] allows to disentangle the high level factors from the intrinsic features of the face using two different encoders that compute the latent representation and the attribute information from the image.

4.4 From CT scans to meshes

In this section we assume that we have one 3D CT scan I_k for each individual k. Computing a mesh of the hip bone from a CT image (Fig. 4.1) is carried out in six steps:

- *(i)* The scans are registered to a common space using the groupwise registration algorithm FROG [3], that provides a transformation field t_k (for each k) that relates the common space to the I_k's image space.

- *(ii)* Each scan I_k is warped according to t_k (so as to obtain I_k in the common

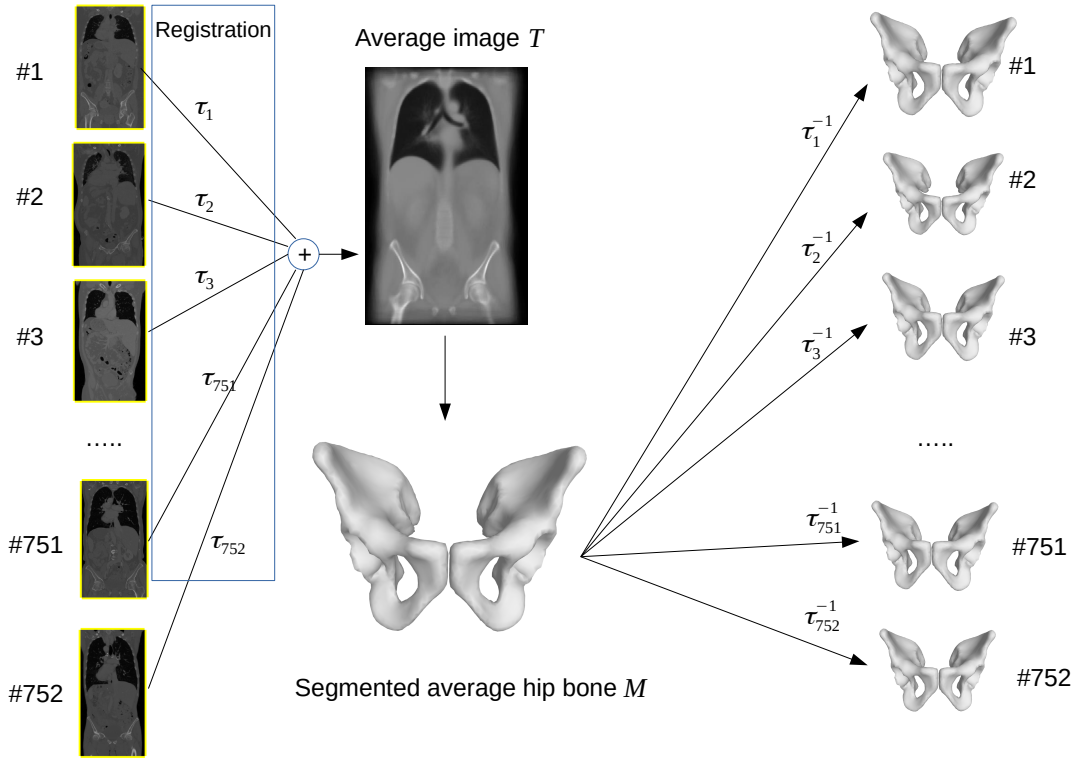


Fig. 4.1 From CT scans to hip bone meshes

space), and a template T is obtained by averaging the warped images.

- *(iii)* The coxal bone is segmented and meshed in T, thus providing a mesh M. The mesh is composed of about 5000 vertices (we denote by P the 3-D points associated with the mesh M).

- *(iv)* The points P are back-transformed in the native space of each scan I_k using the inverse transform t_k^{-1}, providing for each scan I_k a matrix X_k of size $N_p \times 3$ (N_p is the number of points). Each row of X_k is the 3-D coordinate of one point. Note that the points are ordered since the i-th row of each matrix is associated with the same "anatomical" point.

- *(v)* A shape description invariant to position, size and orientation denoted P_k is obtained using a Procrustes alignment of X_k onto P (for each X_k, we estimate a similarity transformation, namely the combination of a rigid transformation with an isotropic scaling transform). A shape description invariant to position and orientation is required since all subjects do not have the same position during acquisition. However, a description invariant to size is more debatable.

- *(vi)* Since the point sets P_k and P are ordered, the mesh M_k is straightforwardly derived from M and P_k.

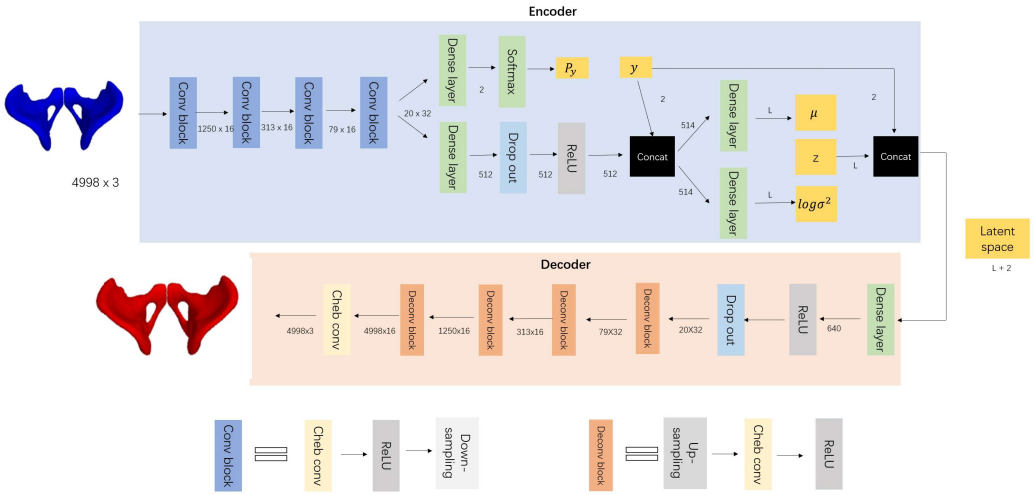


Fig. 4.2 DVAE for sex determination. There are four main steps. **1.** The distribution $q_\phi(y|x)$ (Eq. 4.5) is computed using the neural network q_0 (Eq. 4.6) that outputs the vector P_y whose i-th element is equal to $q_\phi(y = i|x)$ (i = 1 or 2). Then, y is set to the most likely label for testing, and is assumed to be known for training. **2.** The parameters μ and $\log \sigma^2$ (both are vectors of size L) of the distribution $q_\phi(z|x, y)$ (Eq. 4.4) are estimated using the neural networks q_1 and q_2 (Eq. 4.7). The networks q_1 and q_2 share all their layers except the last one. Moreover, q_0 shares with q_1 (and q_2) the four first convolution blocks of the encoder. Note also that y is injected into the networks q_1 and q_2 through a concatenation layer located before the two dense layers. Since one-hot encoding is used to model y, y is of dimension 2. This explains why the concatenation layer takes as input a vector of dimension 512 and outputs a vector of size 514. **3.** For learning, z is sampled from the distribution $q_\phi(z|x, y)$ using the reparameterization trick (Eq. 4.8). For testing, z is set to μ. The latent representation of the input data is composed of y and z and is of dimension $L + 2$. **4.** The reconstruction can be performed from the latent representation using the decoder (Eq. 4.9). Note that the two latent representations $(z, y = \text{"man"})$, $(z, y = \text{"woman"})$ correspond to the "same" individual but of opposite sex. Consequently, by setting y to the man (resp. woman) label in the latent representation, we can reconstruct the original data as a man (resp. woman). This will enable us to transform a sample from a given class into the "same" sample but of another class (see Sec. 4.5.3).

4.5 Disentangled Variational Auto-Encoders for classification and reconstruction

4.5.1 Conditional dependency structure

The proposed model is part of the family of partially-specified models because an explicit latent variable is defined (the sex of the subject) whereas the semantics of the other latent variables is undefined. Several conditional dependency structures can be defined. As an example, [291] explicitly conditions the latent variables z on age c, such that the conditional distribution $p(z|c)$ captures an age-specific prior on latent representations. We propose here to use a conditional dependency structure, as presented in [219, 122],

which is suited to our problem.

We denote by x a sample (a mesh), by y its class (male or female), and by $z \in \mathbb{R}^L$ the other latent variables. Note that the latent representation of x is the pair (y, z). We use the following factorization for the generative process:

$$p_\theta(x, y, z) = p_\theta(x|y, z)p(y)p(z), \tag{4.1}$$

where a weak prior is defined over z and y : $p(z) = \mathcal{N}(z|0, I)$ and $p(y) = \frac{1}{2}$. $p_\theta(x|y, z)$ is modelled as a Gaussian distribution whose mean is given by a neural network f with parameter θ that takes as input y and z. We have:

$$\begin{aligned} p_\theta(x|y, z; \theta) &= \mathcal{N}\left(x \mid f(y, z; \theta), vI\right), \\ &= \mathcal{N}\left(x \mid \widehat{x}, vI\right), \end{aligned} \tag{4.2}$$

where $v > 0$ is a hyperparameter and $\widehat{x}$ is the reconstruction computed from y and z.

As usual in variational inference, the posterior $p_\theta(y, z|x)$ is approximated by $q_\phi(y, z|x)$. In order to disentangle the label y from the other latent variables z, we use the following factorization:

$$q_\phi(y, z|x) = q_\phi(y|x)q_\phi(z|x, y). \tag{4.3}$$

The distribution $q_\phi(z|x, y)$ shows that the estimation of z requires the data x, but also the label y. To understand why this is relevant, let us consider a toy example where z is supposed to represent the size of the subject. If the sex label y is well disentangled from z, z ought to be an intrinsic measure of a subject's size. This means that its estimation needs to regress out the influence of the label y: indeed, a woman who is 160 centimeters tall can be considered as average height while a man of the same height can be considered as short, so that the value of z associated with this woman has to be larger than the one related to this man (even if they have both the same height). Consequently, in order to obtain a disentangled representation, it seems appropriate that z depends both on x and y.

The distribution $q_\phi(z|x, y)$ in Eq. 4.3 is defined as a Gaussian distribution whose mean (resp. covariance matrix) is given by a neural network q_1 (resp. q_2) with parameter ϕ_1 (resp. ϕ_2) that both take as input x and y:

$$q_\phi(z|x, y) = \mathcal{N}(z; \mu, \sigma^2), \tag{4.4}$$

where μ and $\log \sigma^2$ are vectors of size L (see Eq. 4.7 for details). Finally, the distribution $q_\phi(y|x)$ that also appears in Eq. 4.3 is simply defined as:

$$q_\phi(y|x) = Discrete(y|q_0(x; \phi_0)), \tag{4.5}$$

where q_0 is a neural network with parameter ϕ_0 that takes x as input. The output of this network is a positive vector P_y (Eq. 4.6) of size 2 summing to 1: the probability $q_\phi(y = i|x)$ is the i-th element of $q_0(x; \phi_0)$ (i = 1 or 2).

The proposed approach can be summarized as follows:

- If y is known, the neural network q_0 is not required. Otherwise, it acts like a classifier such that the distribution $q_\phi(y|x)$ (Eq. 4.5) is computed as follows:

$$P_y = q_0(x; \phi_0), \tag{4.6}$$

  and y is set to the most likely label.

- The latent variable z is computed from x and y. Firstly, μ and $\log \sigma^2$ that appear in Eq. 4.4 are computed such as:

$$\mu = q_1(x, y; \phi_1), \log \sigma^2 = q_2(x, y; \phi_2). \tag{4.7}$$

  Then, the latent variable z is set to μ for testing new data whereas Eq. 4.8:

$$z = \mu + \sigma \odot \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, I), \tag{4.8}$$

  represents the reparameterization trick that is used for learning (please see the next section). Note that the latent representation of x contains both the variables y and z.

- The reconstruction $\widehat{x}$ can be obtained from y and z as follows:

$$\widehat{x} = f(z, y; \theta). \tag{4.9}$$

The neural networks q_0 (Eq. 4.6), $q1$ and $q2$ (Eq. 4.7) represent the encoder and f is the decoder (Eq. 4.9).

The proposed architecture is depicted in Fig. 4.2. Networks q_0, q_1, q_2 and f (Fig. 4.2) are defined using a combination of the convolutions, max-pooling (downsampling) and upsampling operators presented in [198]. Note that mesh convolution is performed in the spectral domain with a kernel parametrized as a Chebyshev polynomial of order K (K is set to 6).

4.5.2 Parameter optimization

As usual for learning a VAE, the parameters of the DVAE are set to maximize the Evidence Lower BOund (ELBO) [124]. We can show that the term $q_\phi(y|x)$ does not contribute to the loss function because all labels y are known during training. Thus,

maximizing the ELBO does not allow the estimation of ϕ_0 (Eq. 4.6). Consequently, following [219, 122], we add a classification loss $\alpha \log q_\phi(y|x)$ to the ELBO term. The criterion writes:

$$E_{z\sim q_\phi(z|x,y)}\left[\log \frac{p_\theta\left(x,y,z\right)}{q_\phi(z \mid x,y)}\right] + \alpha \log q_\phi(y|x). \tag{4.10}$$

Based on the conditional dependency structure of the model, Eq. 4.10 can be simplified as:

$$\begin{array}{lll} E_{z\sim q_\phi(z|x,y)}\left[log(p(z)) - log(q_\phi(z|x,y))\right] & + \\ E_{z\sim q_\phi(z|x,y)}\left[\log(p_\theta(x|y,z))\right] & + \\ \log(p(y)) + \alpha \log q_\phi(y|x). & \end{array} \tag{4.11}$$

The first term may be expressed as a Kullback–Leibler divergence $(-KL((q_\phi(z|x,y)||p(z)))$ which can be computed analytically since the encoder model and prior are Gaussian. The second term is approximated by a Monte Carlo estimate: we use the SGVB estimator and the reparameterization trick [124] (Eq. 4.8). The third term corresponds to the prior of the label y, that has been set to 1/2. Finally, the last term is computed by the neural network q_0.

The loss function contains two hyperparameters: α that weights the contribution of the classification loss, and the variance v (Eq. 4.2), which is used to compute the second term of Eq. 4.11. As in the VAE case, the variance v weights the contribution of the mean squared error reconstruction and special care is needed to set v. In the following, the two hyperparameters v and α are estimated using cross-validation strategies (note that the influence of the parameter α is limited and could simply be set to 1).

4.5.3 DVAE for classification and reconstruction

The proposed generative model can be used for classification but it also offers the opportunity to transform a sample from a given class to the "same" sample but belonging to another class, by modifying the value of the categorical variables y in the latent representation. The reconstruction of a male mesh (resp. female) as a female mesh (resp. male) is carried out according to the following "sex change" procedure:

- Step 1: The latent variable z is computed from the input data x and its true label y using Eq. 4.7 (z is set to μ). The latent representation corresponds to variables z and y.
- Step 2: We change the value of y in the latent representation, so that we obtain the latent representation of the "same" individual but of the opposite sex.
- Step 3: The reconstruction can be performed with Eq. 4.9 (using the modified latent representation).

In order to test the consistency of the results, we also developed a sex preservation procedure. This is the same procedure as the sex change procedure except that the value of y is not modified in the latent representation (Step 2 is not performed).

Note that the computation of the latent variable z requires knowledge of the sex of the mesh under analysis since the true label y is required to compute μ (Eq. 4.7). For testing, since the sex of the mesh under analysis is not known, we have to replace the true label by its most likely estimate computed with q_0.

However, for the reconstruction step (Eq. 4.9), note that we can choose to reconstruct a subject either as a man or as a woman by setting y in the latent representation appropriately.

4.6 Experiments

Our database consists of 752 CT scans from the University Hospital of Saint-Etienne, France, of which 470 subjects are men and 282 subjects are women. The men are on average 65.8 years old with a standard deviation of 14.2 years and the women are on average 65.6 years old with a standard deviation of 14.6 years.

For each scan, a hip bone mesh is extracted as explained in Sec. 4.4. Each point coordinate is normalized so as to have zero-mean and unit-variance. The means and standard deviations are computed using the training dataset (see Section 4.6.1.2).

In addition to training a DVAE, we also train a vanilla VAE whose architecture is the same as that represented in Fig. 4.2 except that the label y and the computation of P_y (Eq. 4.6) are removed. The usual criterion [124] is used for training the VAE.

We also learn a classifier (denoted C) whose architecture is derived from the one in Fig. 4.2 by keeping only the layers that are useful for the computation of P_y (Eq. 4.6). C and q_0 have the same architecture but q_0 is only a subpart of the DVAE (q_0 shares some layers with q_1 and q_2) whereas C is an independent classifier. The binary cross entropy loss is used for training C.

Finally, we use PyTorch for implementation.

4.6.1 Evaluation protocol

4.6.1.1 Hyperparameter setting

In the VAE case, the variance v is estimated automatically during the training process with the method proposed in [209]: v is computed for each batch as the MSE loss.

Regarding the DVAE, several methods have been tested without success to estimate v automatically. This is why the parameter v as well as the parameter α (Eq. 4.11) are set using cross-validation strategies.

It has been observed that the size of the latent space has limited influence on classification accuracy and on the disentanglement properties for a large range of values of L (for $L = 1$ to 64). However, using too small values of L leads to an increase in the reconstruction error. L has been set to 16 in all experiments. For a fair comparison, the size of the latent space of the VAE has been set to L+2=18.

Optimization of the parameters was done using the Adam optimization algorithm with a batch size of 16. During training, all models are trained for 600 epochs. We keep the same learning rate of 0.0006 for the first 200 epochs and then decay the learning rate to 0.0003 for the next 200 epochs. For the last 200 epochs, we set the learning rate to 0.0001. Training time for DVAE is about 7.2 sec per epoch with a 2080 Ti graphics card. The DVAE needs about 0.2 seconds to generate both male and female hip bones during testing.

4.6.1.2 Nested-cross validation strategy

In order to estimate the ability of the models to handle unseen data and to set the hyperparameters α and v for the DVAE, we follow the nested cross-validation strategy.

First, an (outer) stratified 5-fold cross-validation strategy is used to assess the performance of the models. At each iteration, all folds except one are used as training data (it will be denoted TR) and the remaining one is used as testing data (TE). The three models (DVAE, C, and VAE) are trained from TR and their performances are evaluated on TE. Note that a score can be computed for each fold. We can then derive an average score and its standard deviation.

However, the DVAE learning process requires the hyperparameters α and v to be defined. An inner K-fold cross-validation could be applied at each iteration of the outer cross-validation. However, this would require training a very large number of models. To make the problem tractable, we instead randomly divide the training set TR into a validation set denoted V and a training set T (20% and 80 %). Afterwards, several models are trained from T based on different values for the hyperparameters: a grid search is performed for α and v (α and $\sqrt{v}$ take resp. their value in $\{0.5, 1, 2, 3, 4, 5\}$ and in $\{0.7, 1, 1.3, 1.6, 1.9\}$). Once all models have been trained, the set V is used to select the model that provides the highest disentanglement, that is, the one that leads to the highest success rate for the sex change procedure (see sec. 4.6.1.3). Then, a final model is trained from TR based on the hyperparameters that have led to obtain the selected model (note that TE is used neither to estimate the parameters of the model nor to estimate the hyperparameters).

Table 4.1 Results (mean and standard deviation) obtained with the DVAE approach. CA, OSRSR, SSRSR, and RE stand resp. for classification accuracy, opposite sex reconstruction success rate, same sex reconstruction success rate, and reconstruction error.

CA	OSRSR	SSRSR	RE
$99.59 \pm 0.34\%$	$99.10 \pm 0.92\%$	100%	$1.647mm \pm 0.098mm$

Table 4.2 Comparison with previous works on sex determination. Note that previous works rely on manual estimation (such as lengths, angles or landmark positions) while our approach is fully automatic.

Method	individuals	variables	accuracy
CADOES [55]	256	40 (manual)	97 %
DSP [165, 26]	2040	17 (manual)	$>$ 99 %
Nikita et al. [172]	132	3 (manual)	97 %
Ours	752	5000 (autom.)	$>$ 99 %

4.6.1.3 Evaluation metrics

In the (semi)-supervised case, evaluating disentanglement is often achieved by visualising the reconstructions while modifying the value of a latent variable of interest. In our specific case, this can be easily achieved since the latent variable of interest y is binary (a hip bone is either associated with a man or a woman). Consequently, the model is tested on its ability to perform conditional generation according to the sex label (Sec. 4.6.2.1 proposes quantitative results while Sec. 4.6.2.2 presents some visual examples). The model is also tested for its ability to classify hip bones and to reconstruct the original data.

For each fold, we compute four different metrics to evaluate the performance of the model:

- The classification accuracy (CA) obtained with q_0 (DVAE) or with classifier C.
- The opposite sex reconstruction success rate (OSRSR): we reconstruct a male (resp. female) as a female (resp. male) mesh using the sex change procedure (Sec. 4.5.3). This procedure is considered as successful if the transformed mesh is classified as female (resp. male) using C. This rate should be high if the sex label y has been been properly disentangled from z.
- The same sex reconstruction success rate (SSRSR): we reconstruct a male (resp. female) as a male (resp. female) mesh using the sex preservation procedure (Sec. 4.5.3). This procedure is deemed as successful if the transformed mesh is classified by classifier C as male (resp. female).
- The reconstruction error (RE) in millimeters. The reconstruction obtained with

the sex preservation procedure is compared with the initial mesh in the native space of the image I_k (see Sec. 4.4). The mean of the euclidean distances between each associated point is computed leading to a score for a given subject. This score is then averaged over all subjects in the fold. Note that obtaining the reconstruction in the space of I_k requires the inversion of the normalization step applied to each point coordinate (second paragraph of Sec. 4.6) as well as the similarity transformation (point (v) in Section 4.4).

Note that all metrics except CA are computed using different reconstructions of the mesh under analysis. In order to distinguish between classification errors and reconstruction/disentanglement errors, the true label is used to compute the latent representation.

4.6.2 Experimental performance analysis

4.6.2.1 Quantitative results

The results obtained with the DVAE approach are shown in Tab. 5.1. Regarding the classification accuracy, the DVAE classifier achieves a very high prediction accuracy ($99.59 \pm 0.34\%$). This corresponds to a total of 3 misclassifications out of 752 (one misclassification in 3 folds and zero in 2 folds). The independent classifier C achieves similar results since only three subjects are misclassified (these are not the same subjects).

As a comparison, Tab. 4.2 gives sex prediction accuracy for recent works that are based on the manual positioning of a few landmarks. We cannot claim that the proposed method provides better results since all the methods should be compared on the same database (which unfortunately is not available). However, the proposed method yields state-of-the-art classification results while being free of any manual positioning of landmarks. Moreover, the method is data-driven and not guided by expert knowledge. It is also suited to sex determination from other bones and, as shown in Sec. 4.8.2, from bone fragments.

In terms of reconstruction error, the DVAE performs similarly to a vanilla VAE, which obtains a mean reconstruction error of 1.728 mm, even if the selected values of v at each fold (DVAE) are always larger than those estimated (for each batch) with the method of [209] (VAE). The selected values of v in the DVAE case are relatively large because it has been observed that small values of v lead to poor disentanglement properties. However, an increase in v did not increase reconstruction error.

One could remark that the comparison of the reconstruction errors may be unfair since the true sex label is employed to perform the reconstruction in the DVAE case. However, the same result is obtained when using the estimated label: there are only 3

misclassified cases and using the true label or the false one leads to reconstructions that are mostly similar, except in some specific regions.

Finally, excellent results are obtained for the opposite sex reconstruction success rate, and for the same sex reconstruction success rate. The reconstruction as a female (resp. male) mesh of a male (resp. female) mesh is well-classified by C in more than 99% of the cases (OSRSR). Moreover, the reconstruction as a male (resp. female) mesh of a male (resp. female) mesh is always well-classified by C in our experiments (SSRSR). Note that the accuracy of the classifier C reaches only $97.17 \pm 1.05\%$ when classifying data reconstructed with the vanilla VAE (instead of 100% in the DVAE case).

As noted previously, the comparison with the VAE approach may be unfair since the true label is used for reconstruction in the DVAE case. However, we can use a sex preservation procedure that does not rely on the true label (the label can be estimated by q_0). In this case, when classifying the reconstructions obtained by DVAE, the classifier C reaches an accuracy of $99.59 \pm 0.34\%$, which is exactly the accuracy of q_0 (see Tab. 5.1). Indeed, classifying with C the reconstruction obtained with the DVAE provides exactly the same results than classifying the original mesh with q_0. This clearly shows the consistency of the method. As an example, if a male mesh is considered as a female one by q_0, the DVAE will reconstruct this male mesh as a female one so that the classifier C will be also wrong.

4.6.2.2 Qualitative results

In order to evaluate more precisely the disentanglement properties of the model, each original mesh M_k is compared with its reconstructed (same sex) mesh or with its reconstructed opposite sex mesh. Furthermore, the two reconstructions are also compared together. Note that the two reconstructed meshes are those computed in the previous section (the true label y is used to compute z).

We start by analyzing average results. As in Sec. 4.6.1.3 (please see the definition of RE), the reconstructions (associated with M_k) are computed in the native space of I_k. To compare two (out of the three) meshes, we associate at each vertex v of the template mesh M a real value representing the distance between the two vertices v of the meshes under analysis. These distances are averaged across the different subjects of the testing set. Each vertex of the template mesh therefore receives a color representing the (local) average distance.

These local average distances are represented in Fig. 4.3 (left) when the original meshes are compared with the same sex reconstructed meshes. This comparison shows that the iliac crest is not well reconstructed. This is mainly due to large registration errors that can be observed for some subjects in this region. This makes the problem more difficult because the variability of the data is increased.

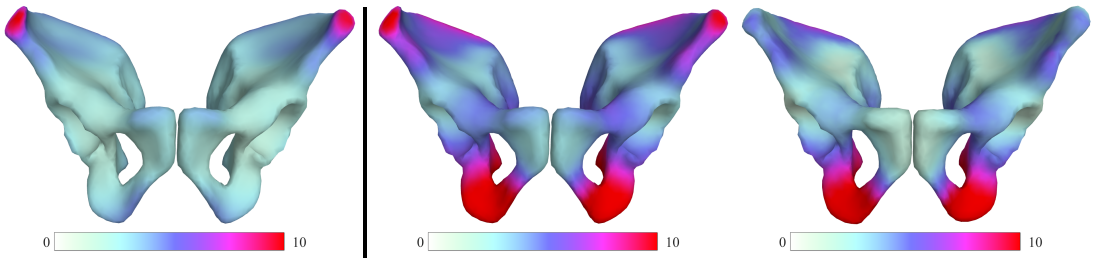


Fig. 4.3 Local average distances. From left to right: original meshes vs reconstructed meshes (lower distance is better), original meshes vs reconstructed opposite sex meshes, reconstructed meshes vs reconstructed opposite sex meshes. Distances are in mm. See text for details.

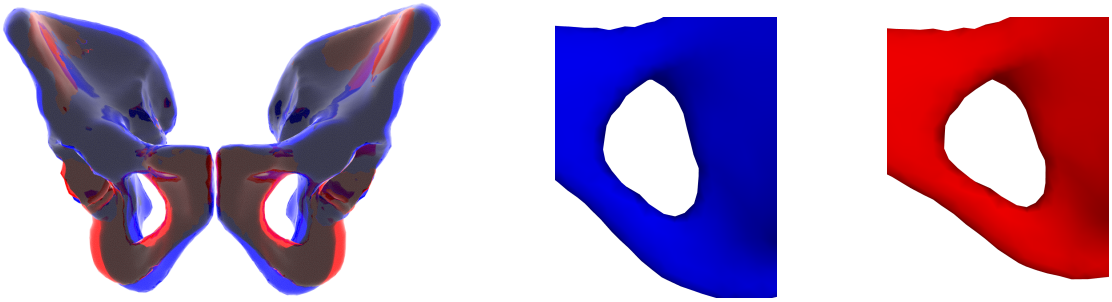


Fig. 4.4 Example of changing a male hip bone (blue) to a female hip bone (red). Left: angle comparison: the subpubic angle is larger for the female bone than for the male bone. Right: the male obturator foramen (left) exhibits an oval shape, while the female obturator foramen (right) exhibits a triangular shape.

As illustrated in Fig. 4.3 (middle) that represents the local average differences between the original meshes and the opposite sex meshes, the opposite sex reconstruction changes the geometry as expected. Moreover, the differences that can be observed are consistent with expert knowledge. As an example, the subpubic angle is known to be larger for women, leading to the difference observed in the pubic arch.

The two reconstructed meshes can be compared (Fig. 4.3 (right)) in order to gain a deeper understanding of the results. This is particularly true for the iliac crest, which is not well reconstructed in both cases. In the case of complete disentanglement of the sex label, we expect this area to be reconstructed similarly for both reconstructions. This is because the iliac crest is known to show little sexual dimorphism compared to other areas of the hip bone. Even if Fig. 4.3 (right) still exhibits differences in the iliac crest between the two reconstructions, they remain low compared to the original reconstruction errors (Fig. 4.3 (left)).

Finally, these results reinforce the idea that the sex variable has been properly disentangled.

We can explore further by analyzing individual results. The analysis of the differences between two meshes was carried out using “cine mode” (rapidly switching

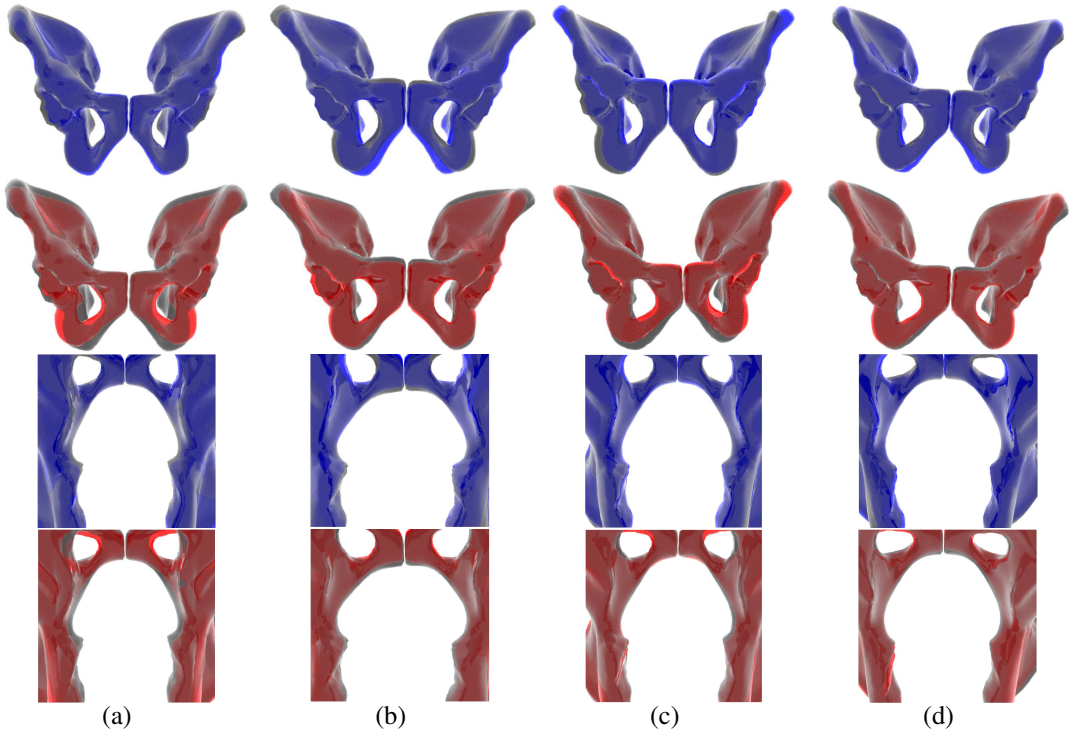


Fig. 4.5 Examples of DVAE results. Original mesh (grey) *vs* mesh reconstructed as a female one (red). Original mesh (grey) *vs* mesh reconstructed as a male one (blue). The original mesh of (b) is a female one while those of (a,c,d) are male meshes.

between them) because the eye is sensitive to movement. For the sake of simplicity, the two meshes are here directly superimposed to compare them (see Fig. 4.4 and 4.5).

When opposite sex reconstruction is successful, the comparison of the opposite sex mesh with the original mesh (or the reconstructed one) reveals the significant anatomical differences between the male and female hip bones, such as the subpubic angle (Fig. 4.4, left) as well as the shape of the obturator foramen (Fig. 4.4, right), of the greater sciatic notch, of the pelvic inlet and of the symphysis. Note that it may sometimes happen that the two meshes do not exhibit all the expected differences, but most of them are generally easily observable.

When opposite sex reconstruction is not successful, the modification is globally consistent, as some significant anatomical differences can be observed, but some of them are sometimes hard to see, or event not present.

4.7 Discussion: In what sense does the method provide understanding?

Predicting sex from a hip mesh is not an easy task for a non-expert and the classification results can be difficult to understand. In the proposed approach, in addition to providing the class of the mesh, its reconstructions as a man and as a woman are also provided. When the original mesh is that of a man (resp. woman), its reconstruction as

a man (resp. woman) is very similar to the original mesh. Conversely, the comparison between the original mesh and its reconstruction with opposite sex exhibits differences in some specific areas (while others remain unchanged). The comparison of these reconstructions with the original mesh enables a non-expert to understand the choice of the classifier, or at the very least to make their own choice.

Fig.4.5(a) gives an illustrative example of the results provided by DVAE. The reconstruction of the mesh as a man is very similar to the original mesh. On the contrary, the reconstruction as a woman exhibits a wider subpic angle and a wider pelvic inlet. Consequently, a non-expert can easily classify the mesh as a male (without using the result of the classifier), or at least, understand why this mesh can be considered as a male one.

It is then legitimate to ask what happens if the label is not correctly estimated by q_0: will the proposed method justify a misclassification or will it detect the mistake? This part should not be considered as a failure case analysis. The purpose of the proposed method is to provide relevant and easily interpretable information so that the users can form their own opinions. Consequently, if the classifier is wrong but the information given by DVAE enables the user to question its decision, this can certainly be considered a positive result.

Both DVAE and C misclassified 3 subjects, we analyze them in detail here. The different reconstructions relative to the misclassified meshes are shown in Fig. 4.5(b,c,d) (note that y is provided by q_0 for the computation of z so as not to bias the results). The 6 misclassified cases can be split into three groups.

The first group is composed of 3 misclassified subjects (one for C and two for DVAE). Fig. 4.5(c) is an illustrative example of this group. It is a man that has been misclassified by C. The reconstruction as a man is very similar to the original mesh in the sex-specific regions, whereas the reconstruction as a woman exhibits some differences in these regions. Consequently, the original mesh seems to be a male mesh and the user may question the choice of the classifier. Moreover, the iliac crest is particularly poorly reconstructed in these 3 subjects. The shape of this region may be responsible for the misclassification.

The second group is composed of 2 misclassified subjects (one for C and DVAE). Fig.4.5(b) is an illustrative example of this group. It represents a woman that has been misclassified by DVAE. When looking at the subpic angles, it seems to be consistent: the reconstruction as a woman is very similar to the original mesh in this area. However, the reconstruction of the pelvic inlet suggests that this is a male mesh (the reconstruction as a man is very similar to the original mesh in this area). Thus, this mesh has both male and female characteristics. This may explain why this subject is difficult to classify. In this case, the two reconstructions enable the user to doubt the result obtained by the

classifier.

The last group is composed of one misclassified subject: this is a man (Fig. 4.5(d)) that has been misclassified by DVAE. When it is reconstructed as a woman, the subpubic angle is slightly increased and the pelvic inlet is made wider, as expected. When it is reconstructed as a man, we expect the reconstruction to be similar to the original mesh but the subpubic angle is slightly decreased. Consequently, the subpubic angle of this man seems to be larger than it should be. This may explain why this subject has been misclassified. However, a user could easily question the results obtained by the classifier, because it seems that the mesh exhibits more male characteristics than female ones.

Finally, the comparison of the two reconstructions with the original mesh is a simple way to understand the choice that was made by the classifier, or to doubt its choice (for the second group) or to question it (for the first and last groups).

4.8 Reconstruction-based classification: application to missing data

4.8.1 Reconstruction-based classification

As written in Sec. 4.7, the comparison of the two reconstructed meshes provided by the DVAE approach with the original mesh enables a non-expert to form an informed opinion. In the same way, one can wonder if the performance of an independent classifier can be improved by feeding the two reconstructed meshes obtained with DVAE to the classifier.

To this end, the following paradigm has been used: after having trained the DVAE, we train an independent classifier denoted C_{recon} whose input data are composed of two meshes: the first one is the original mesh from which we subtract its reconstruction as a man (provided by DVAE, z is computed using the label estimated by q_0) and the second one is the original mesh from which we subtract its reconstruction as a woman. The classifier C_{recon} is identical to C except the first layer that takes an input of size 4998x6 (we have points in R^6 because we model two meshes). In the following, we denote this method DVAE+C_{recon}.

DVAE+C_{recon} achieves an accuracy of 100% for each fold, even with meshes having both female and male characteristics (Sec. 4.7). One possible reason for these results is that the work of C_{recon} is much simpler than the one of C. As an example, let us consider the case of a male mesh. Its reconstruction as a man is very similar to the original mesh so that the first three components of the mesh (we have points in R^6) are close to zero. On the contrary, the reconstruction as a woman exhibits differences in some sex-specific regions so that the last three components of the mesh are close to zero except in the sex-specific regions. Consequently, for a male mesh, all components are

expected to be close to zero except the last three components that lie in the sex-specific regions. For a female mesh, all components are expected to be close to zero except the three first components that lie in the sex-specific regions. By highlighting the regions that allow to distinguish male from female hip bone, the input of C_{recon} is much easier to analyze than the original mesh.

4.8.2 Application to missing data

Since all the classifiers C, C_{recon} and DVAE have already achieved high accuracies, we propose here to make the problem more difficult by introducing missing data: vertices are deleted either on the left-hand, right-hand, lower, upper, front or rear side. The percentage of missing data is expressed in terms of the percentage of the mesh size (in the dimension where the data is removed). As an example, when deleting data on the lower side, the percentage of missing data is expressed in terms of the percentage of the height of the mesh. A very simple imputation strategy is used: missing values are set to the value 0 (which is the mean at each vertex).

Data augmentation is required during training to achieve acceptable results: with a probability of 0.6, the mesh is not modified. Otherwise, it is augmented as follows. The side where the vertices are set to 0 is chosen with a uniform distribution, and the percentage of missing data is selected with a uniform distribution in $0 - 40\%$.

Four different methods are used for classification:

1 The classifier C.

2 DVAE: note that the second term of the loss function (Eq. 4.11) uses the original mesh (and not the augmented one) since we want the reconstruction to be similar to the original mesh.

3 DVAE+ C_{recon}. DVAE is first trained as in the second point. Then, during the learning of C_{recon}, the two reconstructions of an augmented mesh are computed using the DVAE (z is computed using the label estimated by q_0) and the input of C_{recon} corresponds to the augmented mesh from which we subtract its reconstructions. This means that C_{recon} is somehow fed indirectly with augmented meshes during the learning.

4 the last method denoted VAE+C consists in classifying the reconstruction provided by the VAE with C. The VAE is trained in a similar way as the DVAE. Since the VAE provides a reconstruction without any missing data, the classifier C is trained with non-augmented meshes.

Classification accuracy is shown in Fig. 4.6 for a large range of missing data.

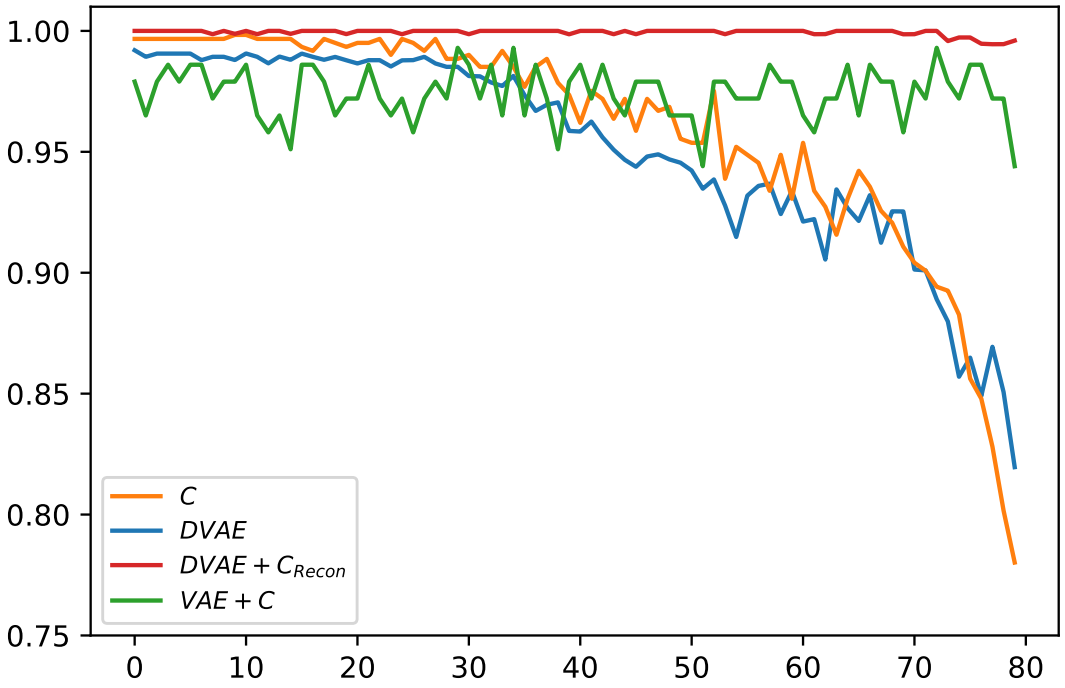


Fig. 4.6 Classification accuracy obtained with different methods in the presence of missing data. The x-axis corresponds to the percentage of missing data ($\times 100$).

As previously, we can note that DVAE and C provide similar results. Even if 70% of the data is missing, C and DVAE can still achieve an accuracy of 90%.

We can also note that the two methods that use reconstructions (VAE+C and DVAE+C$_{recon}$) are quite robust to missing data but DVAE+C$_{recon}$ performs always better than other classification methods. This clearly highlights the benefit of feeding the classifier indirectly with the two reconstructed meshes provided by DVAE.

Finally, the fact that the proposed method is able to achieve very good results in cases where there is a high proportion of missing data seems to indicate that it is able to take into account most of the differences that exist between female and male hip bones.

4.9 Comparison with saliency maps

To compare our approach for the interpretation of mesh classification with a standard method, we have computed SMs for the classifiers C and C$_{recon}$ (without missing data) with the method in [220]. For a given input mesh, the importance w_{ic} at each vertex v_i is computed as follows:

$$w_{ic} = |\frac{\partial p(y=0|x)}{\partial x_{ic}}| = |\frac{\partial p(y=1|x)}{\partial x_{ic}}|, \tag{4.12}$$

where x_{ic} (c =1, 2 or 3) represents either the x, y or z coordinate.

Eq. 4.12 can be computed through back-propagation. For each vertex, the 3 com-

puted importances (one for each coordinate c) are aggregated using the max function: the SM at vertex i is computed as $max_c(w_{ic})$. Instead of considering the derivative of $p(y|x)$, it is also possible to use the unnormalised score (the softmax layer is not considered for the computation of the derivative). In this case, Eq. 4.12 no longer holds and a SM is obtained for each class. Regardless of the methods used or the aggregation function used, the results were always very similar. Fig. 4.7 represents the mean of the SMs (across the subjects), computed with Eq. 4.12 and the max aggregation function.

It is difficult to understand how classifier C makes its decision (Fig. 4.7, left), as the most relevant vertices for the classification are distributed over the entire hip bone (we could expect them to lie specifically in regions that are known to differ between men and women, but this is not the case).

The individual SMs were also extremely different from one another, whereas one would expect that they would all highlight sex-specific regions. Finally, the results were neither intra-architecture repeatable nor inter-architecture repeatable. We suggest that SM may not be suitable for classification problems in which the features that allow distinguishing classes are spatially correlated and scattered. Under these conditions, two classifiers can achieve high accuracy results without having the same decision boundaries, hence their respective SMs will be different.

To illustrate this hypothesis, let us take a simplified problem in which the hip bone is modeled with four variables. To simulate the fact that the hip bone is symmetrical, suppose that x_1 is close to $-x_2$ and that x_3 is close to $-x_4$. The variables x_3 and x_4 represent sex-specific regions ($x_3 \geq 0$ for female hip bones and $x_3 \leq 0$ for male hip bones). Then, let us consider the two following neural networks whose boundary equations are $x_3 - x_4 = 0$ and $x_3\mathbb{1}_{x_1<0} - x_4\mathbb{1}_{x_1\geq 0} + x_1 + x_2 = 0$ (note that $x_1 + x_2$ is likely to be close to 0 for hip bones), where $\mathbb{1}$ is the indicator function. The two neural networks are expected to achieve high accuracy. However, only the SM of the first one is able to highlight the regions of interest x_3 and x_4. The SM of the last one is expected to highlight either x_3 or x_4 according to the value of x_1 as well as two regions that are not sex-specific (x_1 and x_2).

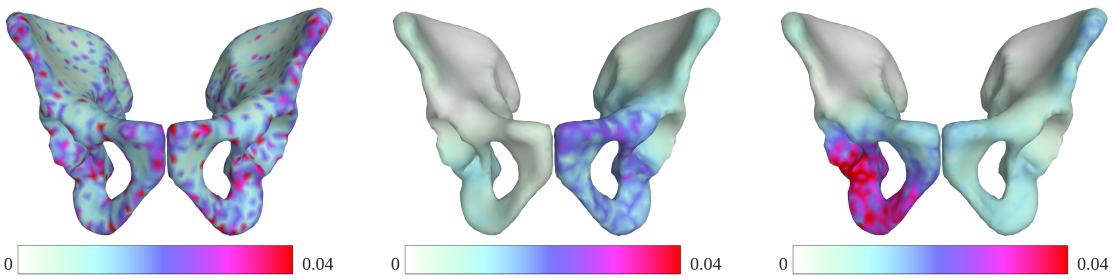


Fig. 4.7 Mean SMs for C (left) and C_{recon} (center and right). The SMs for C_{recon} are either averaged across the female hip bones (center) or the male ones (right).

For C_{recon}, the map is more consistent with our expectations (Fig. 4.7, center and

right) except that a strong asymmetry is observed depending on whether the processed hip bone is a female one or a male one. That is why, the SMs are either averaged across the female hip bones (Fig. 4.7, center) or the male ones (Fig. 4.7, right). Moreover, contrary to the local average distances (Fig. 4.3), the mean SMs highlight the pubic left tubercle, whose shape is known to vary slightly according to the sex (this is clearly visible for the mean SM associated with women, a little less for that associated with men). It seems that the classifier focuses here on a subtle difference between female and male hip bones. Since the input of C_{recon} is partly fed with the output of the DVAE, it can be estimated that this small difference has been captured by DVAE.

Note also that similar mean SMs can be obtained when measuring intra-architecture repeatability and inter-architecture reproductibility. In all cases, the mean SMs associated with men and women highlight a different side of the hip bone and this asymmetry can be more or less pronounced. Moreover, the side of the regions of interest may be permuted: the mean SM of male hip bones highlights the regions that are on the right side (Fig. 4.7, right) but it can be the left side for other tests.

We can conclude that the SMs obtained with C_{recon} are more satisfactory than those obtained with C. Our interpretation is that the input of C_{recon} is much simpler to analyze since the sex-specific regions have been highlighted by DVAE: all components that lie in regions that are not sex-specific are close to 0.

As a final point, it is noteworthy that the proposed method differs significantly from SMs.

First, as written in Sec. 4.2, they do not act at the same level. The SM facilitates understanding of the decision process related to a classification method whereas the proposed approach highlights the differences between the classes and thus provides information on the classification problem to be solved.

Then, an intrinsic limitation of SMs is that they do not provide any semantic meaning on the highlighted regions. In our application, the SMs can at best detect sex-specific regions, i.e. regions that allow to distinguish between male and female hip bones. In contrast, thanks to the conditional generation according to the sex, the proposed method not only provides a sex-specific region detection but also offers the user the opportunity to observe the difference in shape of regions: as an example, we clearly observe with the proposed method that the subpic angle is larger for women (Fig.4.4). Such an approach leads to a better understanding of the class differences.

Moreover, the proposed method provides the users with relevant information so that they can form their own opinions. As an example, we have seen in Sec. 4.7 that the comparison of the two reconstructions enables us to show that some meshes exhibit both male and female characteristics.

Finally, contrary to SMs which is a generic tool, the proposed approach is only

suitable if the label to estimate is a variable corresponding to a source of variability (age, sex, outcomes of genomic-biological-cognitive tests, diagnosis, multicenter variability), which are common situations in medical imaging. As an example, it makes sense in the proposed application to reconstruct a male hip bone as a female one (or a diseased organ into a healthy one) because the latent space can be divided into two independent parts: the non-interpretable part represents the intrinsic (independent of sex) properties of the hip bone and the disentangled part represents the sex label.

4.10 Conclusion

This paper has presented a novel paradigm for the interpretation of classification by neural networks, based on Disentangled VAE representations. The approach provides reconstructions or data generation for each class, which paves the way for a better understanding of class differences. The approach has been illustrated through the interpretation of sex determination from meshed hip bones. It compares favorably with existing methods such as SMs.

The proposed paradigm is comprehensive and suited to the disentanglement and classification of other factors of general interest in medical imaging, such as age, pathology or acquisition parameters. Moreover, there are some cases where some features can be associated with high-level factors. As an example, features related to the disease label may be its severity, and more generally characteristics that model how the disease has transformed the disease-free sample. Note that studies [111, 300] have shown the benefit of modeling both the high-level factors and their related features to disentangle the high-level factors.

Future directions of this work include the modeling of these features and the comparison of the proposed approach with generative adversarial networks that also can achieve disentanglement in a supervised setting. Moreover, learning the significant differences between the classes (at the population level) during training is another perspective that would help to determine if the differences observed for a particular sample under classification are related to opposite sex reconstruction or if they stem from other reasons such as registration inaccuracy. This may further help the analysis of the results.

CHAPTER 5

Joint disentanglement of labels and their features with VAE

In Chapter 4, we discuss the classification and generation capabilities of the DVAE (Disentangled Variational Autoencoder). However, relying solely on a single value to represent a label may not fully capture its associated characteristics. While this may not be immediately apparent in the task presented in Chapter 4, it becomes more evident with attributes such as eyeglasses. If we simply encode the label value (1/0) into the latent space, we lose control over the generation process, as we cannot determine the specific type of glasses that will be generated.

To learn the representation of features associated with labels, our goal is to learn a disentangled representation which consists of two parts: the identity information and the label feature representation. To validate the effectiveness of the disentangled representation learning, we conduct the feature swapping task, such as transferring eyeglasses from one image to another. Previous approaches learn the disentangled representation and perform this task in different ways.

As discussed in Sec. 2.2, CCVAE [111] is suitable for this task. It builds upon the VAE framework and incorporates an extra autoencoder specifically for the labels, learning a representation of the label and connecting it with the latent representation of the data. However, it often exhibits suboptimal generation quality and unsatisfactory generation accuracy.

Another option is to use GANs for this task. As presented in Sec. 4.7, ELEGANT [265] swaps the attributes associated with the labels by manipulating the latent encodings as shown in Fig.2.16. However, GAN training can be unstable and may result in artifacts in the generated images.

In this chapter, we propose an alternative factorization approach within the VAE framework that incorporates the label feature u. Our latent space is composed of three components: the identity information z, the label variable y, and the feature associated with the label u. This formulation allows us to manipulate the high-level factors of the images by modifying the latent space. You can find hereafter an article titled "Joint Disentanglement of Labels and Their Features with VAE," which was accepted for publication at the 2022 IEEE International Conference on Image Processing (ICIP).

The authors of the article are Kaifeng Zou, Sylvain Faisan, Fabrice Heitz, and Sebastien Valette.

To improve both the generation quality and accuracy, we further extend this approach by utilizing VQVAE (Vector Quantized Variational Autoencoder) to learn a discrete latent space. Additionally, we introduce a two-step learning procedure to ensure stability during training. Through experimental results, we demonstrate that these strategies contribute to achieving excellent generation quality and robust disentanglement of the desired features. This approach is presented in Chapter 6, which primarily includes an article titled "Disentangling high-level factors and their features with Conditional Vector Quantized VAEs." This article has been accepted by Pattern Recognition Letters and authored by Kaifeng Zou, Sylvain Faisan, Fabrice Heitz, and Sébastien Valette. Additionally, we also provide the supplementary material in this chapter which contains a sensitivity analysis and a comparison with ELEGANT [265], which has not been published previously.

5.1 Abstract

Most of previous semi-supervised methods that seek to obtain disentangled representations using variational autoencoders divide the latent representation into two components: the non-interpretable part and the disentangled part that explicitly models the factors of interest. With such models, features associated with high-level factors are not explicitly modeled, and they can either be lost, or at best entangled in the other latent variables, thus leading to bad disentanglement properties. To address this problem, we propose a novel conditional dependency structure where both the labels and their features belong to the latent space. We show using the CelebA dataset that the proposed model can learn meaningful representations, and we provide quantitative and qualitative comparisons with other approaches that show the effectiveness of the proposed method.

5.2 Introduction

It is a key challenge to learn disentangled representations where variables of interest are independently and explicitly encoded [18]. These representations allow to manipulate data by modifying high level factors (e.g. removing or adding glasses to a person's face). Probabilistic generative models, such as Variational Autoencoders (VAE) [124] are popular to learn such representations in the unsupervised [30, 86, 118], (semi-)supervised [219, 122], and in the weakly-supervised [207] cases. We focus hereafter on the semi-supervised case because supervision yields better disentangled models [152].

Most previous works [158, 219, 122] divide the latent representation into two components: the non-interpretable part and the disentangled part corresponding to variables

that explicitly model the factors of interest. Each factor of interest is therefore associated to a latent variable of the same type. As an example, if the label of interest refers to the glasses (1 when the subject is wearing glasses, 0 otherwise), there will be a categorical variable in the latent space that encodes the presence or absence of glasses. However, this variable does not allow to model the features of the glasses (e.g. shape/size/color of the glasses), that can be either lost, or at best entangled in the other latent variables.

To our knowledge, only [111] proposed to address this problem. In [111], a feature is associated with each high level factor. Moreover, the latent space no longer contains the labels but their features (the label is used to condition its associated feature). We propose here a novel conditional dependency structure that allows to model both the labels and their features. Contrary to [111], the latent space contains in the proposed model both the labels and their features. Finally, we use an original architecture to build the decoder of the VAE. We show that AdaIN [99] improves the quality of the reconstructed images and that the use of learnable tokens [44, 125] improves disentanglement properties of the model.

Finally, note that generative adversarial networks can also be used to obtain disentangled representations: the methods proposed in [265] and [264] also allow to manipulate the features related to high level factors. This is achieved by swapping attributes between pairs of images. However, these methods are only able to accomplish a small number of the tasks that can be performed with VAE-based methods. As [265] and [264], the proposed method can also swap the high level factors and the related features of two images. However, (i) it allows also to generate new images by sampling from the model (without any other input or with high level factors only), (ii) it allows also to modify the high level factors and the associated features for a single image (by sampling), (iii) it provides finally a classifier that estimates the high level factors. Note also that the methods of [265] and [264] are fully supervised whereas the proposed method handles arbitrary supervision rates.

5.3 Disentanglement of labels and their features from other latent variables

5.3.1 Conditional dependency structure

For the sake of simplicity, we consider here that a unique label (high level factor) is provided for an image. The extension to several labels is straightforward. For the illustration, we consider the binary case where the label is 1 (e.g. if the subject is wearing glasses), or 0 otherwise.

Let x be an image, y its label, u the features related to label y, and z the other latent

variables that are supposed to carry no information on y and u. The latent space is formally composed of y, z and u. The generative process is inspired by the previous work of [122], except that no feature u is defined in [122]. It writes:

$$p_\theta(x, y, z, u) = p_\theta(x|y, z, u)p(u|y)p(y)p(z), \tag{5.1}$$

where θ stands for the parameters of the decoder. A weak prior is defined over z and y: z follows a zero-centered multivariate normal distribution with unit variance ($p(z) = \mathcal{N}(z; 0, I)$) and y follows a uniform discrete distribution. $p_\theta(x|y, z, u)$ is modelled as a Gaussian distribution whose mean is computed by a neural network (the decoder d_θ of parameter θ) that takes as input y, z and u. We have:

$$p_\theta(x|y, z, u) = \mathcal{N}(x; d_\theta(y, z, u), vI), \tag{5.2}$$

where v is a hyperparameter. Finally, a special care is needed to model $p(u|y)$. In our application, the feature vector u encodes the shape/size/color of the glasses. So as to favor disentanglement properties of the model, the two prior distributions (one for each possible value of y) differ from each other. Two different approaches denoted as PA1 and PA2 (proposed approach 1 and 2) are considered:

- Case $y = 1$ (glasses). For both approaches, $p(u|y = 1)$ is a zero-centered multivariate normal distribution with unit variance.
- Case $y = 0$ (no glasses). For PA1, $p(u|y = 0)$ is a multidimensional Dirac delta function, enforcing the components of u to be zero. For PA2, it is a zero-centered multivariate normal distribution with a variance equal to the identity matrix multiplied by 0.1, favoring the components of u to be close to 0.

PA1 seems to be a better choice since images with no glasses should all have the same value of u. We use PA2 to show that the proposed modeling may work with a less informative prior.

The posterior $p_\theta(y, z, u|x)$ is approximated by $q_\phi(y, z, u|x)$ which can be factorized as:

$$q_\phi(y, z, u|x) = q_\phi(y|x)q_\phi(z|x, y)q_\phi(u|x, y), \tag{5.3}$$

where ϕ stands for parameters of the encoder. In Eq. 5.3, we assume that z and u are independent conditionally to x and y. The distribution $q_\phi(y|x)$ is a discrete distribution whose probabilities are provided by the softmax layer (See Fig. 5.1). The distribution $q_\phi(z|x, y)$ is defined as a Gaussian distribution whose mean (resp. covariance matrix) is given by the encoder. For PA1 (case y=1 only) and for PA2, the distribution $q_\phi(u|x, y)$ is defined in the same way as $q_\phi(z|x, y)$. For PA1 (case y=0), $q_\phi(u|x, y = 0)$ is modeled

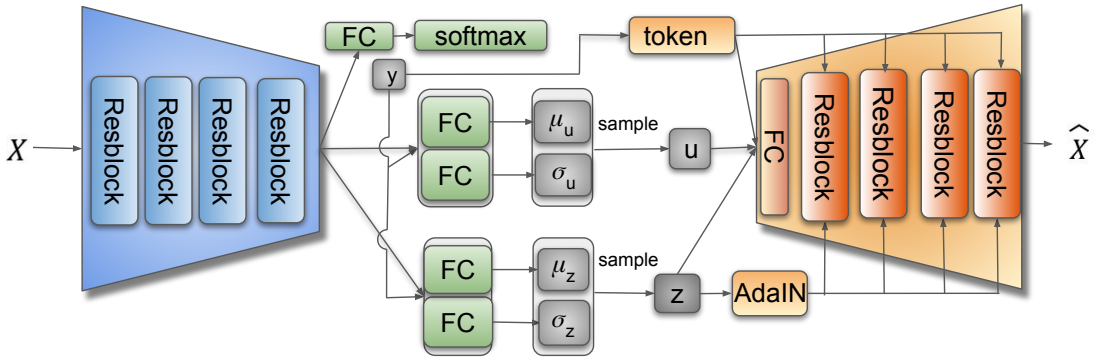


Fig. 5.1 Model architecture. FC stands for fully connected layer. For testing, if y is not known, y is set to the most likely label (based on the output of the softmax layer that represents $q_\phi(y|x)$). z and u are set to μ_z and μ_u. For training (Section 5.3.2), if y is not known, y is sampled from $q_\phi(y|x)$ using a Gumbel-softmax relaxation. z and u are sampled from $q_\phi(z|x,y)$ and $q_\phi(u|x,y)$.

as a multidimensional dirac delta function. Indeed, in this case, the prior distribution $p(u|y=0)$ tells us that u is the null vector.

The proposed architecture is depicted in Fig. 5.1. We use AdaIN [99] as a normalization method: it enables the information carried by z to be transferred to each layer of the decoder. AdaIN injects the latent variable z to each layer of the decoder through a fully connected layer that is not shown in Fig. 5.1. Moreover, we use one set of learnable tokens [44, 125] per class. The set is then selected according to the value of y. Each set is composed of five tokens (one scalar and four images that are associated each one to a residual block of the decoder). The first one (the scalar) is concatenated to u and z to feed the first fully connected layer of the decoder. Then, for each token (an image), we concatenate the token and the input of its associated residual block along the channel dimension. It allows the information provided by y to be transferred to each input of the residual block.

Finally, for PA1, u is multiplied by y. It enables to constrain u to be a null vector if y is 0, and not to modify its value otherwise ($y=1$).

5.3.2 Parameter optimization

If y is known, the optimization of $\log p(x,y)$ can be achieved by maximizing the ELBO (Evidence Lower BOund), that writes: $E_{z,u\sim q_\phi(z,u|x,y)} log(p_\theta(x,y,u,z)/q_\phi(z,u|x,y))$. As in [122], we add a classification loss $\alpha \log q_\phi(y|x)$ to the ELBO term. By using Eq.

6.1 and 5.3, we obtain the following criterion (it is divided by the number of pixels N):

$$\begin{array}{ll} \beta E_{z\sim q_\phi(z|x,y)}\left[\log(p(z)) - \log(q_\phi(z|x,y))\right] & + \\ \beta E_{u\sim q_\phi(u|x,y)}\left[\log(p(u|y)) - \log(q_\phi(u|x,y))\right] & + \\ \frac{1}{N} E_{z,u\sim q_\phi(z,u|x,y)}\left[\log(p_\theta(x|y,z,u))\right] & + \\ \beta \log(p(y)) + \alpha \log q_\phi(y|x) & \end{array} \tag{5.4}$$

The two first terms are a Kullback–Leibler divergence which can be computed analytically since the distributions are Gaussian except for the second term in the case of PA1 with y=0. In this case, it vanishes to 0 since both distributions are equal. Then, the third term is approximated by a Monte Carlo estimate: we use the SGVB estimator and the reparameterization trick [124] (with the notation of Fig. 5.1, we have: $(z,u) = (\mu_z,\mu_u) + (\sigma_z,\sigma_u) \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0,I)$). The fourth term corresponds to the prior of the label y, that has been set to 1/2. Without loss of generality, the variance v (Eq. 6.3) is set to 1 to compute the third term of Eq. 6.6 and the other terms of the ELBO are weigthed by a factor β. Consequently, two hyperparameters have to be set: α and β.

If y is not known (semi-supervised case), it has to be treated as a latent variable. Marginalization can be performed [122]. We sample y, as in [219], from the discrete distribution $q_\phi(y|x)$ using a Gumbel-softmax relaxation.

5.4 Experiment

We experiment on the CelebA dataset [150], with an image size of 128x128. The glasses label has been selected because it leaves little room for subjectivity. The hyperparameters of the methods have been set by using a cross-validation strategy on the training set. Concerning the criterion (Eq. 6.6), α has been set to 1 (it has little influence on the results) and β to $1e-4$. Since the second term of the ELBO (for y=1) leads to degrade the results, its weight (for y=1) has been divided by 100 (for both PA1 and PA2). We use the Adam optimizer [123] with a learning rate equal to $1e-4$ and a batch size of 32. The sizes of z and u are set to 100 and 16 respectively. The supervision rate has been set to 0.2.

We compare our method with CCVAE [111] and with the model M2 of [122]. Two different architectures are used for CCVAE. We first use the implementation of the authors. Since it is adapted to the processing of images of size 64x64, the sizes of the input/output layers have been modified accordingly. This method is denoted as CCVAE. For the second method denoted as CCVAE2, we adapt the architecture of our model to the conditional dependency structure of CCVAE. As an example, y is no longer part of the latent space in CCVAE so that it is no more used for estimating the reconstruction.

Table 5.1 Quantitatives results in terms of (i) success rates for removing and adding glasses (SR(-) and SR(+)), (ii) LPIPS between the original images and the reconstructed ones, and (iii) balanced classification accuracy (BCA).

Model	SR(-)	SR(+)	LPIPS	BCA
CCVAE	**99.38%**	19.37%	0.4414	95.45%
CCVAE2	95.52%	47.68%	0.2549	94.26%
M2	80.34%	33.03%	0.2564	96.13%
PA1	96.31%	59.85%	0.2484	96.55%
PA2	94.98%	**64.25%**	**0.2416**	**97.09%**

In the same way, we adapt the architecture of the proposed model to the conditional dependency structure of M2. It leads to the removal of u from the modeling. For all models, the size of the latent space that models the attributes of the face has been set to 100.

As mentioned in the introduction section, the VAE-based methods allow to accomplish several tasks. For the sake of simplicity, we will only consider three different tasks for comparison purposes: the classification task, the reconstruction task, and the exchange of high level factors and of the related features (if they exist) between two images. The latter task enables us to clearly observe the disentanglement capability of the model. Indeed, it enables to check that the model disentangles not only the label but also the features (of the glasses) from the face attributes z. To evaluate the quality of the reconstructed images, we use Learned Perceptual Image Patch Similarity (LPIPS) [287] that computes perceptual difference between two images. The disentangled ability of the model is evaluated by computing the success rate of swapping. To this end, we select random pairs of images composed of one image with glasses (I_g) and one image without (I_w). Their values of y and u are then exchanged. We consider that the glasses are correctly removed from I_g (resp. added to I_w) if the reconstruction (after the attribute swapping) is classified as $y = 0$ (resp. $y = 1$) with an independent classifier based on ResNet 50. We denote by SR(-) (resp. SR(+)) the success rates for removing (resp. adding) glasses. Results obtained with the different approaches are shown in Tab. 5.1. Since SR(+) is not a perfect evaluation criterion for measuring disentanglement properties of the models (it does not check that the glasses added to I_w are those of I_g), Fig. 5.2 presents swapping results for 6 pairs of images in the case of PA1 (PA2 provides similar results), M2 and CCVAE2.

First, all methods obtain good classification accuracy (BCA) despite a supervision rate equal to 0.2.

Regarding the quality of the generated images (LPIPS), all methods, except CCVAE achieve very similar results. This is mainly due to the fact that the decoders of all the methods (except CCVAE) are very similar. Moreover, we observed that the removal of AdaIN leads to a substantial increase of LPIPS (without modifying significantly the

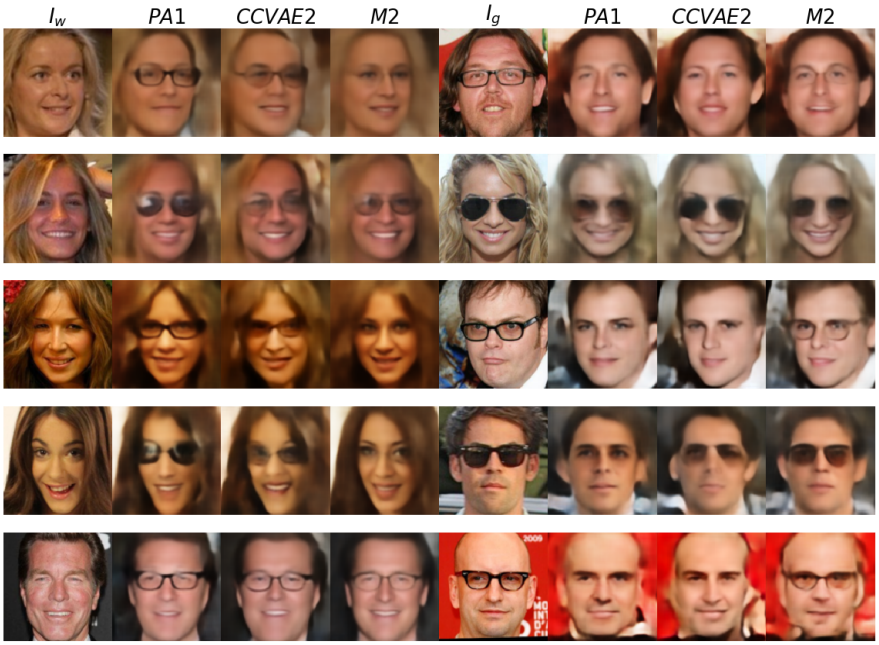


Fig. 5.2 Attribute swapping for 5 pairs of images using PA1, CCVAE2 and M2. For each line, the second, third and fourth image should be I_w with the glasses of I_g. The three rightmost images should be I_g, but without the glasses.

other evaluation criteria). As an example, the removal of AdaIN for PA1 brings the LPIPS criterion from 0.2484 to 0.3059. This clearly highlights the benefit of AdaIN: it allows to improve the reconstruction of the images by transferring to each layer of the decoder information carried by z.

With respect to the success rates of swapping (SR(-) and SR(+)), results obtained with M2 are not very satisfactory, thus illustrating the importance to model the features related to the label. Since the glasses (for M2) are actually well-reconstructed without any label/feature swapping, their features may be entangled in the other variables z of the latent space. This makes the addition of glasses difficult because modifying y is not enough: other variables of the latent space have to be modified to define some proper features of the glasses to be added. Conversely, it appears that the removal of the glasses is simpler (SR(-)>SR(+)) insofar as modifying the label y is enough.

For CCVAE, results obtained for SR(+) and SR(-) do not inform us about the disentanglement properties of the model because the generated images are actually so blurred that it is most of the time difficult to observe the glasses.

Results obtained with CCVAE2 are better than those obtained with M2 in terms of SR(+) and SR(-), illustrating the interest of modeling the features related to a label. However, we observe Fig. 5.2 that the features of glasses cannot be transferred to other images. This means that two images with the same values of u do not exhibit the same glasses. Since the modification of u still leads to the modify the features of the glasses, the features of the glasses are partially entangled in the other latent variables

Fig. 5.3 Multiple attribute swapping with our method. We add to the 3 images of the first column the beard associated with the image which is located on the same line on the rightmost and the glasses associated with the images of the first line.

with CCVAE2.

Finally, the proposed approaches (PA1 and PA2) achieve a success rate for adding glasses that is superior to those obtained with other models as well as a very high success rate for removing glasses. Moreover, we can observe (Fig. 6.8) that the proposed model (PA1) correctly extracts the features of the glasses from the image I_g and is able to reconstruct them reasonably well on another image, which shows that the label as well as the features of the glasses have been properly disentangled from the attributes of the faces. Similar results are obtained for PA2. Note that the results presented in Fig. 5.2 cannot be considered as representative: $SR(+)$ is about 60 % for PA1 but PA1 obtains good results for all pairs of images of Fig 5.2. The proposed methods achieve actually very good results (the glasses added to I_w match those of I_g and the glasses are correctly removed from I_g) for many pairs of images. However, such results are extremely rare with CCVAE2 and M2. These results show the relevance of the proposed conditional dependency structure, and in particular the benefit of y being in the latent space. We have also noted that the tokens favor the disentanglement properties of the model by allowing the information provided by y to be transferred to each input of the residual block of the decoder. As an example, for PA1, the removal of the tokens (y is then just used as an input of the fully connected layer of the decoder) brings SR(+) down from 59.85% to 47.23% (SR(-) is not modified significantly).

Finally, the proposed method can easily be extended to the case of several high level factors. In the case of two factors, (y_1, u_1) and (y_2, u_2) can be considered as inde-

pendent for the generative process. Then, for the variational approximation, we write $q_\phi(y_1, y_2, z, u_1, u_2|x)$ as $q_\phi(y_1|x)q_\phi(y_2|x)q_\phi(z|x, y_1, y_2)q_\phi(u_1|x, y_1)q_\phi(u_2|x, y_2)$. Results obtained with the glass and the beard labels are shown in Fig. 6.8. They illustrate that the proposed model allows to manipulate the attributes of beard and glasses separately.

5.5 Conclusion

The proposed approach compares favorably to other VAE-based approaches, thus showing the interest of modeling both the labels and their features in the latent space. Moreover, our experiments illustrate the benefit of using AdaIn and learnable tokens to build the decoder: the first one allows to improve the quality of the generated images while the second one favors disentanglement properties of the model. To further improve the quality of the generated images, a perspective of this work could be to replace the Gaussian prior on z by a categorical distribution [246]. Better reconstruction may also favor a better disentanglement.

CHAPTER 6

Disentangling high-level factors and their features with Conditional Vector Quantized VAEs

6.1 Abstract

Two recent works have shown the benefit of modeling both high-level factors and their related features to learn disentangled representations with variational autoencoders (VAE). We propose here a novel VAE-based approach that follows this principle. Inspired by conditional VAE, the features are no longer treated as random variables over which integration must be performed. Instead, they are deterministically computed from the input data using a neural network whose parameters can be estimated jointly with those of the decoder and of the encoder. Moreover, the quality of the generated images has been improved by using discrete latent variables and a two-step learning procedure, which makes it possible to increase the size of the latent space without altering the disentanglement properties of the model. Results obtained on two different datasets validate the proposed approach that achieves better performance than the two aforementioned works in terms of disentanglement, while providing higher quality images.

6.2 Introduction

There is a key challenge to learn disentangled representations where high-level factors would be independently and explicitly encoded [18]. Disentangled representations allow to manipulate data by modifying high level factors, thus paving the way to easier interpretation of the influence of these factors [301]. It has also been shown that these representations may be more sample-efficient, less sensitive to nuisance variables, and better in terms of generalization [247]. They are thus used in many applications such as face attribute manipulation [83], action generation [182] and image-to-image translation [132].

There is a substantial literature on disentangled representation learning [148]. Since better disentangled models can be obtained under supervision [152], we are only interested in the (semi)-supervised case, and specifically in Variational Autoencoder meth-

ods (VAE). VAEs [124] are versatile models of choice to learn such representations in the semi-supervised case [219, 122]. To achieve disentanglement with a VAE, the latent representation is generally divided into two parts [158, 219, 122]: the non-interpretable part and the disentangled part corresponding to variables that explicitly model the factors of interest. But these variables only represent the labels associated with the factors of interest and not the features that can be related to these factors. Consequently, these features are either lost, or entangled in the other latent variables. The works of [111, 300] clearly show that modeling both labels and their associated features improves the model's disentanglement properties. In [111], a feature is associated with each high level factor. The latent space is composed of two different sets of random variables: the first one is composed of features associated with the labels, and the second one models information not directly associated with any of the labels. This implies that the latent space no longer contains the labels, but each label is used to condition its associated feature (in the latent space). Subsequently, this method will be denoted CCVAE (characteristic capturing VAE).

In [300], we proposed a novel conditional dependency structure where both the labels and their features belong to the latent space. In this model, the conditional priors of the features given the label have to be set properly to ensure the desired disentanglement properties. Moreover, the loss function is composed of two Kullback-Leibler divergences (KLD), that have to be weighted differently, so as to achieve satisfactory results. This makes the approach [300] difficult to use. This second method will be denoted JDVAE (Joint disentanglement of labels and their features with VAE) in the following.

In this article, we propose, as in [111, 300], a VAE-based approach that models explicitly both the high-level factors and their associated features. The proposed model will be denoted CVQVAE (Conditional Vector Quantized VAE), and can be considered as an extension of the work of [300]. To overcome the limitations of [300], the features are no longer considered as random variables over which integration has to be performed. Instead, each feature is here (deterministically) computed from the input data using a neural network whose parameters can be estimated jointly with those of the decoder and of the encoder. These features (as well as the labels and the latent variables) are then used by the decoder to reconstruct the data. This approach is inspired by conditional VAE (CVAE) [226, 270, 34], except that the conditioning variable is known for CVAE, and computed in CVQVAE. We thus obtain a simplified model (free of conditional priors for the features, and a single KLD loss). Moreover, to improve the quality of the generated images and in particular to generate less blurry images, the Gaussian prior on the latent representation has been replaced by a categorical distribution [246]. The resulting model is more difficult to optimize, but we circumvent this problem with

an efficient two-step learning procedure. The proposed model outperforms the two approaches mentioned above on two different datasets.

6.3 Conditional Vector Quantized Variational AutoEncoder

6.3.1 Architecture of the model

Without loss of generality, we consider for the presentation of the model that there is one binary high-level factor (label y). Note that the extension to several high-level factors is straightforward. The architecture of CVQVAE is illustrated in Fig. 6.1. The underlying latent representation of the image x is composed of the label y, along with the (other) latent variables z. Finally, c denotes the (continuous) features related to y. As an example, for face images, the "glasses" label y is equal to 1 if the subject is wearing glasses, 0 otherwise. c represents the (continuous) features of the glasses (shape/size/color) and z models the intrinsic properties of the face.

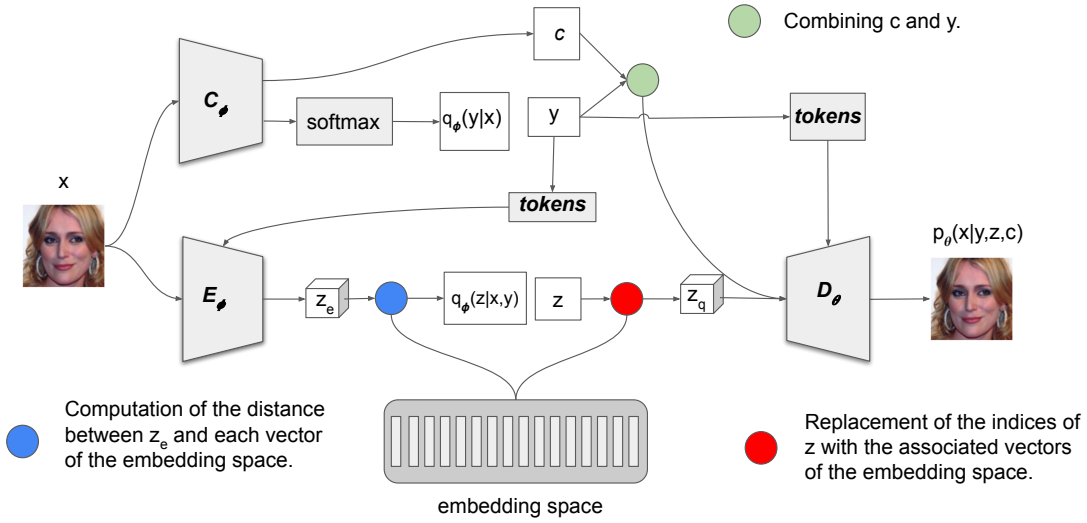


Fig. 6.1 Architecture of CVQVAE. E_ϕ consists of 5 residual blocks. C_ϕ consists of 5 residual blocks followed by one single fully connected layer. D_θ is composed of one fully connected layer followed by 5 residual blocks.

As shown in Fig. 6.1, the proposed model is composed of an encoder (E_ϕ and C_ϕ), a decoder (D_θ), an embedding space and tokens (ϕ and θ refer to the parameters of the encoder and of the decoder). It relies on the estimation of distributions $q_\phi(y|x)$, $q_\phi(z|x,y)$ and $p_\theta(x|y,z,c)$. Sec. 6.3.2 explains the reasoning behind this choice and how the distributions are defined. Finally, all the parameters of the model are jointly estimated (Sec. 6.3.3).

The encoder: It is composed of two neural networks E_ϕ and C_ϕ: C_ϕ takes as input x and outputs the features c and the label distribution $q_\phi(y|x)$. Then, y is set to the most likely label for testing. When training (semi-supervised case), it is set to its true

value (if y is known), or sampled from $q_\phi(y|x)$. Finally, E_ϕ takes as input x and y (as tokens) and outputs z_e which is used in conjunction with the embedding space to compute $q_\phi(z|x,y)$. z is either sampled from this distribution during training (See Sec. 6.3.3) or set to the most likely value during testing.

The embedding space: As in [246], an embedding space, composed of K vectors of R^D, is used to model the categorical distribution $q_\phi(z|x,y)$ (see Sec. 6.3.2). Moreover, the indices of z are replaced with the vectors of the embedding space (of the same indices) to obtain z_q.

The decoder: The decoder D_θ outputs the distribution $p_\theta(x|y,z,c)$. Under the Gaussian assumption of Eq. 6.3, this is achieved by outputting the mean of this distribution. As shown in Fig. 6.1, D_θ is not directly fed with z, y and c. A new variable z_q is computed from z (previous paragraph), tokens are used for representing y (next paragraph) and c and y are combined deterministically to feed D_θ (last paragraph).

The tokens: The label y is not directly fed into E_ϕ and D_θ. As in [300, 182], the label information y is encoded through the use of learnable parameters. They are used here to transfer the y label information to each input of the convolution blocks of E_ϕ and D_θ. As in [300, 182], these parameters are called tokens. We have two sets of learnable tokens for E_ϕ that each consist of five images (each image is associated with a residual block of the encoder). The set is selected according to the value of y. For each convolutional residual block, we concatenate the token and the input of the block along the channel dimension. The same strategy is used for the tokens of D_θ. Additionally to the five images, the two sets related to D_θ have another token that is a scalar one: it is concatenated to z_q (z_q is flattened).

Finally, c is not directly fed into the decoder D_θ. D_θ takes as input a feature vector generated by combining y and c deterministically. To enhance model flexibility, the components of this vector only encode information related to one label ($y=0$ or $y=1$): components encoding a property for $y=0$ are zero if $y=1$ or vice versa. This procedure is also adapted to the meaning of the high-level factor. As an example, the two high-level factors, "smile" and "glasses", differ from the fact that the features associated with the "smile" label have a meaning whether the person smiles (y=1) or not (y=0), whereas the features associated with the "glasses" label encode the shape/size/color of the glasses, thus having only a meaning in the case $y=1$ (for $y=0$, there is nothing more to encode than the fact that $y=0$). Considering the "glasses" label, c is multiplied by y. It enables us to constrain the resulting vector to be a null vector if y is 0, and to be equal to c otherwise ($y=1$). For the "smile" label, each label ($y=0$ and $y=1$) has its own features. Consequently, the components of c are divided into two equal parts. The first and the second parts represent respectively features for $y=0$ (neutral face) and for $y=1$ (smiling face). The components of the first part and of the second part

are multiplied by $1 - y$ and y, respectively, so that the first part's components are zero if $y = 1$ and the second part's components are zero if $y = 0$. In the following, N_c refers to the number of components of each part of c: for instance, c is of size N_c for the "glasses" label and of size $2N_c$ for the "smile" label.

6.3.2 Conditional dependency structure

The generative process of CVQVAE is inspired by the work of [122], except that no feature c is defined in [122], and by the CVAE approach [226, 270, 34]. It writes:

$$p_\theta(x, y, z|c) = p_\theta(x|y, z, c)p(y)p(z), \tag{6.1}$$

where θ represents the parameters of the decoder. Following the idea of CVAE, our purpose should be to approximate the posterior $p_\theta(z, y|x, c)$. However, contrary to [226, 270, 34], the value of c is actually not given, but is computed from x with C_ϕ. Since c is deterministically obtained from x, we have: $p_\theta(z, y|x, c) = p_\theta(z, y|x)$. Consequently, we approximate the posterior $p_\theta(z, y|x)$ by $q_\phi(z, y|x)$ where ϕ represents the parameters of the encoder. It writes:

$$q_\phi(z, y|x) = q_\phi(z|x, y)q_\phi(y|x). \tag{6.2}$$

The distributions in Eq. 6.1 and 6.2 are modeled as follows: y follows a uniform discrete distribution. In accordance with [124], $p_\theta(x|y, z, c)$ is modelled as a Gaussian distribution: its mean is computed by a neural network (the decoder D_θ of parameter θ) that takes as input y, z and c. We have:

$$p_\theta(x|y, z, c) = \mathcal{N}(x; D_\theta(y, z, c), vI), \tag{6.3}$$

where v is a hyperparameter. As in [122], $q_\phi(y|x)$ is a discrete distribution whose probabilities are provided by a softmax layer. Instead of using the traditional Gaussian assumption, we follow the idea of [246] to model the prior on z and the distribution $q_\phi(z|x, y)$ so as to improve the quality of the generated images.

In [246], z is a map (of size $N_z \times N_z$) and each component of z is a categorical variable that represents the index of a vector of a shared embedding space (this space is composed of K vectors of R^D). Each component of z is independent and identically distributed and follows a uniform discrete distribution. Moreover, $q_\phi(z_{i,j} = k|x)$ (there is no variable y in [246]) is set to 1 for $k = \arg\min_k ||E_{\phi_{i,j}}(x) - e_k||$, 0 otherwise, where $E_\phi(x)$ is the continuous output of the encoder (and $E_{\phi_{i,j}}(x)$ its value at coordinate (i, j)), and where e_k is the k-th vector of the shared embedding space. Here, we propose to set the posterior $q_\phi(z_{i,j} = k|x, y)$ as a function of $||E_{\phi_{i,j}}(x, y) - e_k||$. The smaller

$||E_{\phi_{i,j}}(x,y) - e_k||$, the larger the probability $q_\phi(z_{i,j} = k|x,y)$ should be. It is defined as:

$$q_\phi(z_{i,j} = k|x,y) = \frac{e^{-||E_{\phi_{i,j}}(x,y)-e_k||}}{\sum_{k=1}^{K} e^{-||E_{\phi_{i,j}}(x,y)-e_k||}}. \tag{6.4}$$

Sec. 6.3.3 explains the relevance of this modeling based on the loss function to be optimized.

6.3.3 Parameter optimization

If y is known, the optimization of $\log p_\theta(x,y|c)$ can be achieved by maximizing the Evidence Lower BOund (ELBO). Under the reasonable assumption that z and c are conditionally independent given x and y, it writes:

$$\log p_\theta(x,y|c) \geq E_{z\sim q_\phi(z|x,y)} log(p_\theta(x,y,z|c)/q_\phi(z|x,y)). \tag{6.5}$$

Note that Eq.6.5 (and Eq. 6.1) are defined for arbitrary values of c. In the proposed approach, since c is set as a function of x, optimization of the ELBO also allows us to estimate c from x and y. By using Eq. 6.1, the ELBO term writes (we drop the constant term $\log p(y)$):

$$E_{z\sim q_\phi(z|x,y)} \log(p_\theta(x|y,z,c)) - KL(q_\phi(z|x,y)||p(z)), \tag{6.6}$$

where KL is the Kullback–Leibler divergence. The first term is approximated by a Monte Carlo estimate: we propose to use the Straight-Through Gumbel-Softmax estimator [106] to sample from $q_\phi(z|x,y)$. Moreover, without loss of generality, the term $\log(p_\theta(x|y,z,c))$ in Eq. 6.6 can be replaced by the mean squared error between x and $D_\theta(y,z,c)$ provided that the second term of Eq. 6.6 is weighted by a factor β.

The second term can be computed analytically since both distributions $q_\phi(z|x,y)$ and $p(z)$ are discrete. This term acts as a regularization term that constrains the latent space to have good properties: close samples in the latent space should have similar reconstructions. In [246], this term cannot play its role because the choice of the distribution $q_\phi(z_{i,j} = k|x)$ leads to a constant KL divergence. Hence we propose a distribution $q_\phi(z_{i,j} = k|x)$ that allows to obtain such a regularization. Under our hypothesis, the term $-KL(q_\phi(z|x,y)||p(z))$ can be obtained by summing over (i,j) the entropy of $q_\phi(z_{i,j}|x,y)$ (up to a constant).

If y is unknown (semi-supervised case), y is sampled from $q_\phi(y|x)$ as in [219] using a Gumbel-softmax relaxation and the same loss function is used.

Finally, in both cases, three additional terms are added to the loss function. As in [122], we add a classification loss $\alpha \log q_\phi(y|x)$ to the ELBO term (α is set to 1) because the term $q_\phi(y|x)$ does not contribute to the loss function if y is known. Moreover, since

a Gumbel-softmax relaxation is used to sample z, the gradients are simply copied from z_q to z_e, similarly to straight-through gradient estimation in [246]. Consequently, the parameters of the embedding space do not receive gradients from the loss and we use the additional term presented in [246] to learn the embedding space. Finally, a commitment loss presented in [246] is also used (its weight is set to 0.25 as in [246]).

6.3.4 Architecture and training variations

To obtain a more detailed evaluation of our contributions, we suggest a range of alternatives, labeled A through F, with our current CVQVAE method denoted as E. Approaches A through D employ standard initialization strategies and the relevant loss function to train the models' parameters, while for approaches E and F, a two-step learning procedure is implemented.

Approaches A and B are based on the proposed CVQVAE except that no feature is associated with y (i.e. c is removed from the model). The resulting models have also the same conditional dependency structure as the model M2 in [122]. The distribution $q_\phi(z|x, y)$ is modeled as proposed in [246] for approach A and as proposed in Sec. 6.3.2 (Eq. 6.4) for approach B.

Approach C corresponds to the proposed CVQVAE with standard training. Approach D is based on the CVQVAE with two differences: instead of using a discrete latent representation for z, z follows a zero-centered multivariate normal distribution with unit variance ($p(z) = \mathcal{N}(z; 0, I)$) and the distribution $q_\phi(z|x, y)$ is defined as a Gaussian distribution whose parameters are given by the encoder [124]. Moreover, as in [300], we use AdaIN [99] as a normalization method. AdaIN injects the latent variable z to each layer of the decoder through a fully connected layer. Using AdaIN causes the decoder to attach greater importance to z. The model associated with approach D is denoted as CGVAE (conditional Gaussian VAE).

Approach E is similar to approach C, relying on the proposed CVQVAE method. Approach F employs a model named CGVAE2, which is similar to CGVAE but without the use of AdaIN. Both approaches use a two-step learning procedure. The rationale behind two-step learning is that the optimization problem would be easier to solve if c was known: to this end, we start to train a simplified model (approach D with a small latent space) that also has the C_ϕ network (that enables us to compute y and c) as well as the tokens. Then, for the estimation of the parameters of CVQVAE (approach E) or of CGVAE2 (approach F), the parameters of the C_ϕ network and the tokens are initialized with those obtained by approach D. Note that these parameters are frozen during the first iterations of the optimization procedure.

6.4 Experiments

Implementation details: We experiment on the CelebA [149] and CheXpert [104] datasets each containing more than 200000 images (80% is used for training) of size 128^2. The first dataset is composed of labeled face images, on which we conduct quantitative experiments (for the "glasses" and "smile" labels), as well as qualitative experiments (for the "beard" and "makeup" labels). The second dataset is composed of labeled X-ray chest images on which three experiments are conducted.

Hyperparameters (N_z, N_c, K, D, β) have been tuned using a cross-validation strategy in the experiment relative to the "glasses" label with CVQVAE. Other experiments use the same tuned hyperparameters: N_c has been set to 16 and β to 1e-4 (see text under Eq. 6.6). When modeling z as a categorical variable, the size of the latent space z has been set to $S_z = N_z \times N_z$ with $N_z = 8$, and the embedding space is composed of $K = 512$ vectors of dimension $D = 16$. For CGVAE2 (approach F), S_z has been set to 1024 which is equal to the number of components of z_q for CVQVAE (1024=8×8×16). This allows for a fair comparison between CVQVAE and CGVAE2. For CGVAE (approach D), we set S_z to a small value (100) to obtain a simplified model with better convergence properties. Note that approach D is mainly useful to initialize CVQVAE and CGVAE2.

The models have been trained independently for each experiment. We used the Adam optimizer with a learning rate equal to 10^{-4}, a batch size of 32 and a supervision rate set to 0.2. The experiments were conducted using PyTorch 1.9 and CUDA 10.2, leveraging a Nvidia 1080Ti graphics card.

Evaluation metrics: We consider two different tasks: the classification task, and the exchange of high level factors and their related features between two images (so as to measure the disentangled properties of the model). The classification task is assessed using the Balanced Classification Accuracy (BCA).

The disentangled ability of the model is evaluated by computing the success rate of swapping the attributes. In order to distinguish between classification errors and disentanglement errors, the true labels are used to perform this task: we select random pairs of images composed of one image of both classes denoted $x_{y=1}$ and $x_{y=0}$. Their values of c and y are then exchanged to create two fake images. They are generated by feeding the decoder with z_0, c_1, $y = 1$ (for the first one), and with z_1, c_0, $y = 0$ (for the second one), where z_0, c_0, and z_1, c_1 denote the latent variables and the features computed from $x_{y=0}$, and $x_{y=1}$, respectively. As an example, for the "glasses" label, the first fake image should exhibit the face of $x_{y=0}$ with the glasses of $x_{y=1}$ and the second fake image should show the face of $x_{y=1}$ without glasses. We consider that the swap ("from 0 to 1" or "from 1 to 0") is successful when the associated generated image is well-classified by an independent classifier based on ResNet 50 [82]. We denote by

Table 6.1 Results for the "glasses" label in terms of (i) success rates of swapping SR(-) and SR(+), (ii) CFD, and (ii) FID that compares the distribution of fake images with the one of real images. The models are described in the text.

Model	SR(-)	SR(+)	CFD	FID
A	91.07%	72.26%	0.200	20.38
B:A+KLD	93.46%	77.46%	0.142	20.03
C:B+c	99.61%	76.91%	0.112	20.78
D:CGVAE	99.99%	62.96%	0.114	21.27
E:CVQVAE	**100%**	**79.13%**	**0.093**	**20.05**
F:CGVAE2	99.85%	72.83%	0.097	20.51

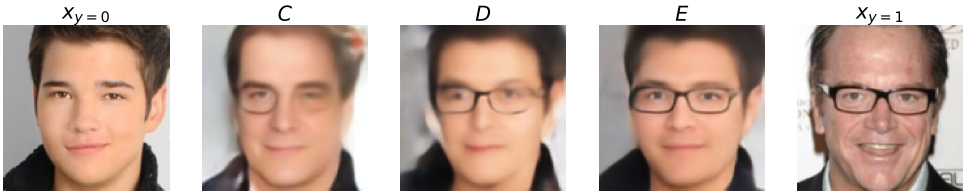


Fig. 6.2 Attribute swapping ("glasses label") using the C, D and E (our) approach. The second, third and fourth images should be $x_{y=0}$ with the glasses of $x_{y=1}$.

SR(+) (resp. SR(-)) the success rates for going from "0 to 1" (resp. "1 to 0").

Note that SR(+) and SR(-) are not perfect evaluation criteria for measuring disentanglement properties of the models. As an example, for the "glasses" label, SR(+) does not check that the glasses added to $x_{y=0}$ are those of $x_{y=1}$. Consequently, some swapping results will be presented to check whether the features are well-transferred or not. Moreover, in order to obtain a quantitative criterion, we propose to compute the Classification Feature Distance (CFD) as the L2 norm between two outputs of the last layer of the independent classifier. These two outputs are obtained by feeding the classifier once with the original image $x_{y=1}$ ($x_{y=0}$, resp.) and once with the fake image that has the same values of c and y as the original image: the fake image is generated by the decoder with z_0, c_1, $y = 1$ (z_1, c_0, $y = 0$, resp.). As a reminder, z_0, c_0, and z_1, c_1 denote the latent variables and the features computed from $x_{y=0}$, and $x_{y=1}$. The CFD is based on the assumption that an ideal attribute swap should not change the features extracted by the classifier. We also compute one Fréchet Inception Distance [85] (FID), that compares the distribution of fake images with the one of real images.

6.4.1 Comparison of approaches A to F

Results obtained with approaches A to F are provided in Tab 6.4 for the "glasses" label. A and B perform well, but they cannot transfer the features of the glasses to another image since glasses are not explicitly modeled. Moreover, the regularization over the latent space, induced by the proposed modeling of $q_\phi(z|x, y)$ (Eq. 6.4), improves the

Table 6.2 Results for the "glasses" and the "smile" labels in terms of CFD, success rates of swapping SR(-) and SR(+), FID, and in terms of BCA.

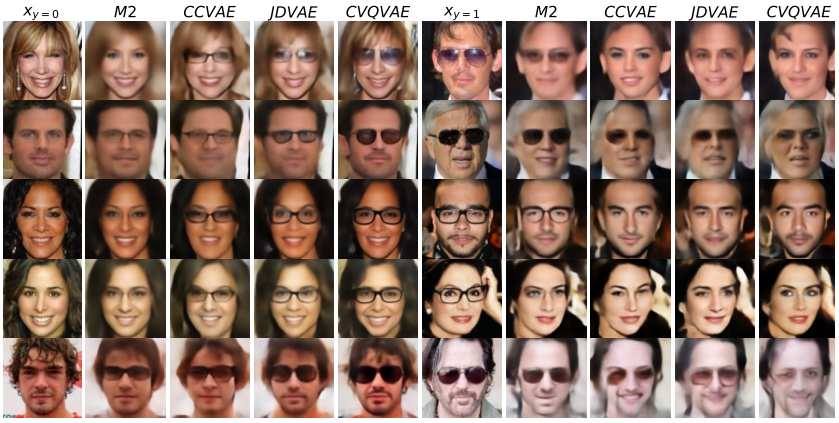
	glasses					smile				
Model	CFD	SR(-)	SR(+)	FID	BCA	CFD	SR(-)	SR(+)	FID	BCA
CCVAE	0.145	95.52%	47.89%	21.10	96.26%	0.052	**96.97%**	74.75%	16.14	**90.80%**
JDVAE	0.098	94.98%	64.25%	21.66	**97.09%**	0.059	81.80%	79.34%	16.36	90.52%
M2	0.274	80.34%	33.02%	22.27	96.13%	0.069	44.07%	50.72%	16.75	90.48%
CVQVAE	**0.093**	**100%**	**79.13%**	**20.05**	96.67%	**0.049**	89.25%	**90.25%**	**14.54**	90.09%

Table 6.3 Results for three different pathologies in terms of CFD, success rates of swapping (SR = (SR(-)+SR(+))/2), FID, and in terms of BCA.

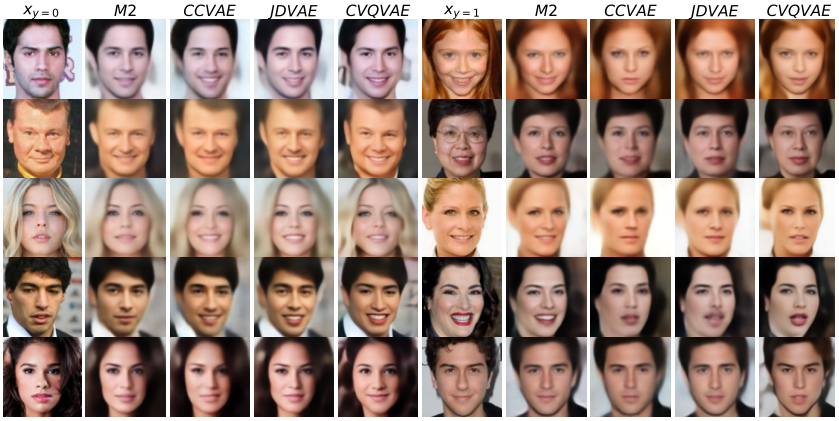
	cardiomegaly				atelectasis				consolidation			
Model	CFD	SR	FID	BCA	CFD	SR	FID	BCA	CFD	SR	FID	BCA
CCVAE	0.298	57.51%	8.01	**80.33%**	0.137	49.38%	7.67	71.92%	0.215	62.16%	7.04	**80.67%**
JDVAE	0.261	60.27%	7.82	79.44%	0.138	50.92%	7.62	71.11%	0.250	62.23%	7.07	80.07%
M2	0.347	47.36%	8.99	79.58%	0.233	41.99%	7.99	70.99%	0.446	36.93%	8.43	80.58%
CVQVAE	**0.169**	**69.97%**	**7.13**	79.92%	**0.134**	**64.55%**	**6.87**	**72.86%**	**0.117**	**72.36%**	**6.91**	80.46%

disentanglement properties of the model: SR(+) and SR(-) obtained with B are larger than those obtained with A. Thanks to the modeling of c, approach C obtains better results in terms of SR(-) and CFD. However, visual inspection of the results show that c not only carries information about the glasses but also about the face, as illustrated in Fig. 6.10 (C): the glasses are well transferred from $x_{y=1}$ to $x_{y=0}$ but some features of the faces are also transferred. The use of AdaIN in approach D results in a slightly deterioration of the model's disentanglement properties ($SR(+)$ decreases), and the modeling of z (the latent space is only 100) leads to a reduction of image quality. However, the modification of c does not change the face anymore (see Fig. 6.10 (D)), thus showing that c is free of any information about the face.

Results obtained with Approach E (CVQVAE) enable to obtain the best results in terms of quantitative criteria (Tab. 6.4). Moreover, visual inspection of the results (Fig. 6.10(E)) shows that the properties of the glasses are relatively well transferred, while preserving the main features of the face. Finally, as in [300, 111], these results clearly illustrate the interest of modeling the features related to the high-level factors. Indeed, as shown by the values of SR(+) and SR(-), CVQVAE yields better disentanglement representations than methods A and B for which the properties of the glasses are not modelled. Note also that AdaIN is not used in the CVQVAE approach. AdaIN was shown in [300] to improve the reconstruction of the images. However, it is no longer worth using AdaIN when the size of the latent space is increased. Furthermore, the use



(a)



(b)

Fig. 6.3 Attribute swapping ("glasses" label (a) and "smile" label (b)) with M2, CCVAE, JDVAE and CVQVAE. For each row, the second, third, fourth and fifth images should be $x_{y=0}$ with the glasses (a) or smile (b) of $x_{y=1}$. The four rightmost images should be $x_{y=1}$, but without glasses (a) or with the neutral attitude of $x_{y=0}$ (b).

of AdaIN slightly weakens the disentanglement properties of the model.

Finally, while CGVAE2 yields very satisfactory results, CVQVAE provides better results than CGVAE2, both in terms of disentanglement and image quality. Moreover, increasing β for CGVAE2 produces disentanglement properties similar to CVQVAE, but at the expense of image quality (they are blurry, data not shown). These results illustrate the relevance of using a discrete latent representation.

6.4.2 Comparison with state-of-the-arts methods

The proposed approach is compared with two VAE-based approaches that also model the features related to the high-level factors: CCVAE [111] and JDVAE [300] and with the model M2 of [122] with a Gaussian prior for z (the features are not modelled). Finally, for all methods, the architectures of the encoder and of the decoder are similar to those of JDVAE[300]. For these approaches, the size of the latent space S_z has been

set to 100: larger latent space results in a model that is more difficult to optimize and leads to a reduction in the performance model.

Results obtained with the "glasses" and with the "smile" labels are provided in Tab. 6.2 and in Fig. 6.3. First, all methods obtain good classification accuracy (BCA) despite a supervision rate equal to 0.2. Note that the accuracy of Resnet 50 is 98.69% and 93.07% for the "glasses" and the "smile" labels, respectively. With respect to the quality of the fake images (FID), CVQVAE provides images of better quality, thus justifying the use of a larger latent space. Regarding the success rates of swapping (SR(-) and SR(+)), results obtained with M2 are less satisfactory than those obtained with the other methods, showing once again the interest of modeling both the high-level factors and their features. An analysis of the results obtained by CCVAE, JDVAE, and CVQVAE for SR(+) and SR(-) requires to consider the labels separately. For the "glasses" label, results obtained with CCVAE are relatively satisfactory but the features of glasses are not well-transferred (CFD values in Tab. 6.2 and results in Fig. 6.3). Results are more satisfactory with JDVAE [300]. However, CVQVAE obtains the best success rates for adding and removing glasses. Additionally, our method correctly extracts most of the features of the glasses from the image $x_{y=1}$ and reconstructs them reasonably well on $x_{y=0}$ (Fig. 6.3), which shows that the label and features of the glasses have been properly disentangled from the attributes of the faces.

For the "smile" label, visual inspection of the reconstructed images (without attribute swapping, data not shown) shows that JDVAE and CCVAE have difficulties in extracting the features related to the smile. As an example, for a neutral face with open mouth, its reconstruction shows a closed mouth. Similarly, for a smiling face with wide open mouth, the mouths of the reconstructed images are less open. On the opposite, CVQVAE provides better reconstructions. Our hypothesis is that the problem is made easier with CVQVAE because the components of c that represent the neutral face are not the same than those representing the smiling face. Regarding the success rates of swapping (SR(-) and SR(+)), results obtained with CCVAE look satisfactory, especially for SR(-) but this number is biased. SR(-) is actually greater than the accuracy of ResNet 50 (when classifying neutral face). This shows that it is easier for the classifier to classify neutral fake images than real neutral images. This is due to the fact that the neutral images obtained with CCVAE are actually too neutral. Indeed, we can observe that the features related to the smile are not properly transferred to other images (see Fig. 6.3). As we saw previously, this is not only a feature transfer problem, but also a feature extraction problem. Results are actually slightly improved with JDVAE [300], but the best results are undoubtedly obtained with CVQVAE.

Fig. 6.4 shows results obtained with other labels, which further illustrate the versatility of the model.

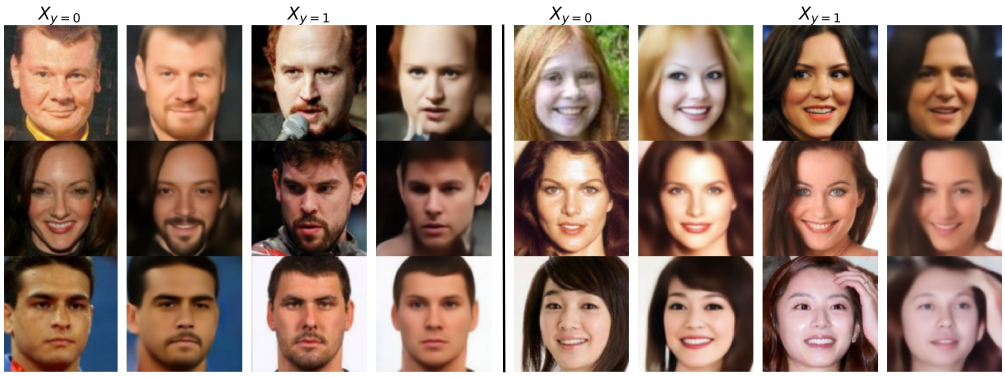


Fig. 6.4 Attribute swapping for "beard" (left) and "makeup" labels (right). Presentation is similar to Fig. 6.3 except only results of CVQVAE are shown.

In addition, we show the effectiveness of our model on the CheXpert dataset (Tab. 6.3). Three different experiments have been conducted. In these experiments, $y = 1$ is associated to a pathology ("cardiomegaly", "atelectasis" or "consolidation") and $y = 0$ is related to the "non finding" label (no pathology). Quantitative results show once again that CVQVAE outperforms the other methods. Since no feature has been related to the label $y = 0$ (it was also the case for the "glasses" label), it is possible to reconstruct an image with a pathology as an image without pathologies. The difference between its reconstruction and its reconstruction as a "free of pathology" image reveals the influence of the pathology (in green on Fig. 6.5).

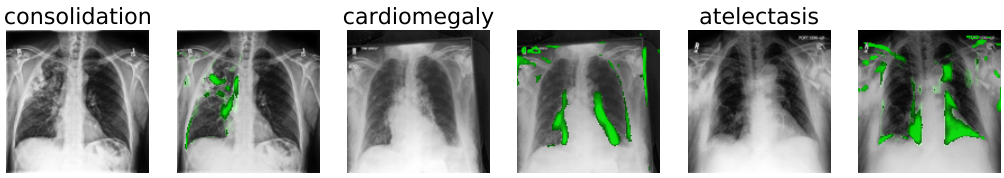


Fig. 6.5 Results obtained with three different pathologies on the CheXpert dataset. For each pathology, the original image (with the name of the pathology at its top) is on the left, and the regions in green (at the right of the original image) represent regions that differ the most between the reconstruction and the "pathology-free" reconstruction.

6.4.3 Exploration in the feature space c

We have also carried out several experiments on the information encoded by the variable c. In Fig. 6.6, fake images are generated by feeding the decoder with z_1, y_1 and $c = c1 + \alpha(c2 - c1)$ ($\alpha \in [0, 1]$), where z_i, y_i, and c_i denote the variables related to image x_i (i=1 or 2) with y_1=y_2. As an example, for the "glasses" label, if $y_1 = 1$, the generated glasses should be similar to those of x_1 (if α is close to 0), of x_2 (if α is close to 1), or in-between (for other values of α). Moreover, in all cases, the generated face should be the one of x_1. Results shown in Fig. 6.6 are consistent with our expectations: interpolation

in the feature space c results in a smooth transition between smiles (top), neutral faces (middle), or types of glasses (bottom).

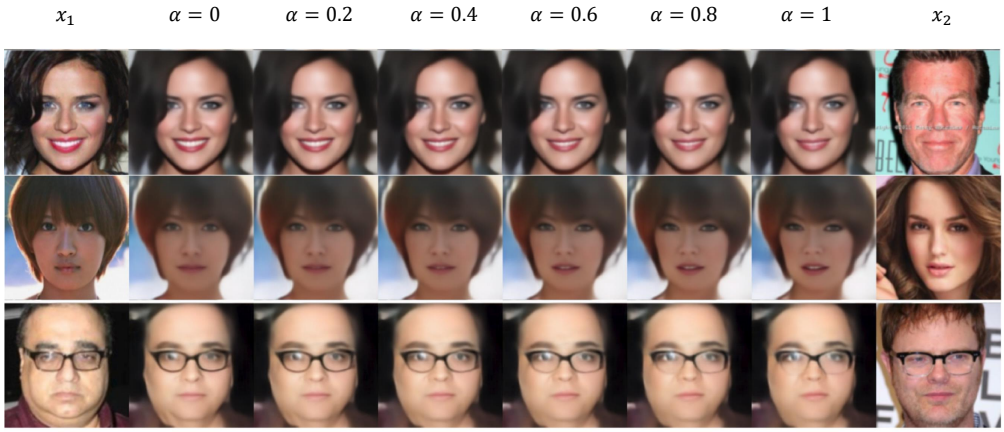


Fig. 6.6 Interpolation in the feature space with different values of α using x_1 and x_2 (see text for details). Each column corresponds to the generated results for α given at its top, with the exception of the left and right columns that correspond to x_1 and x_2. The bottom row is related to the "glasses" label (with $y_1 = 1$) while the two other rows correspond to the "smile" label ($y_1 = 1$ for the top row and $y_1 = 0$ for the middle one).

In Fig. 6.7, the influence of the magnitude of c is shown: images are generated by feeding the decoder with z, y and λc ($\lambda \in [0, 2]$), where z, y, and c are computed from x. Results are shown for the "smile" label for $y = 1$ (Fig. 6.7, top) and for the "glasses" label for $y = 1$ (Fig. 6.7, bottom).

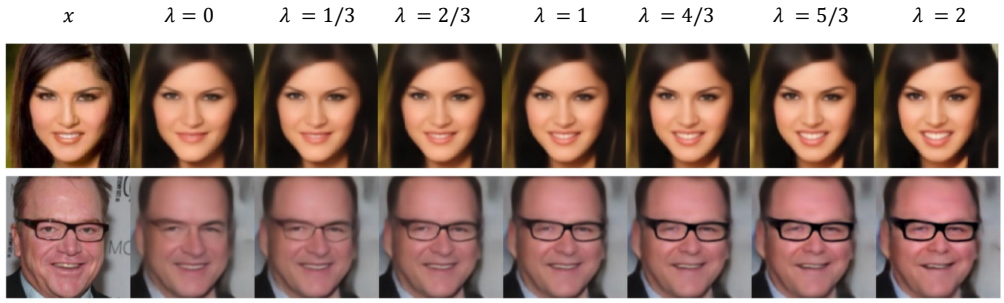


Fig. 6.7 Increasing or decreasing the magnitude of c from x with different values of λ (see text for details). Each column corresponds to the generated results for λ given at its top, with the exception of the left column that corresponds to x. The top row is related to the "smile" label (with $y = 1$) while the bottom row corresponds to the "glasses" label (with $y = 1$).

Increasing or decreasing the magnitude of c leads to amplifying or reducing the related features in the generated images. For example, with $\lambda = 2$, the frames of glasses become very dark and wide, and the way of smiling is also exaggerated (the mouth is notably more open). Moreover, even if $y = 1$, a null value for c ($\lambda = 0$) prevents glasses from being generated.

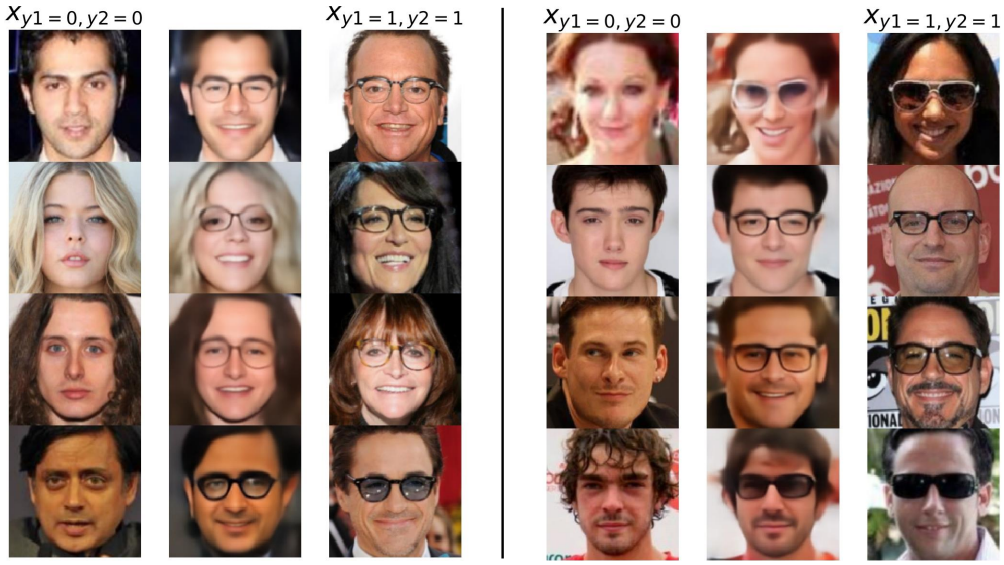


Fig. 6.8 Multiple attributes transfer with CVQVAE. There are 8 different examples, 2 per row, so that there are 4 on the left and 4 on the right. For each example, glasses and smile from $x_{y_1=1,y_2=1}$ are transfered to $x_{y_1=0,y_2=0}$, and the resulting image is located between $x_{y_1=0,y_2=0}$ and $x_{y_1=1,y_2=1}$.

6.4.4 Multiple attribute disentanglement

Our approach can easily be extended to the multiple attribute case. Two high-level factors are considered hereafter: y_1 and y_2 denote the labels, and c_1 and c_2 denote the related features. Equations of Sec. 6.3 remain valid by setting y to (y_1, y_2), and c to (c_1, c_2). We use the following assumption: $p(y) = p(y_1)p(y_2)$ and $q_\phi(y|x) = q_\phi(y_1|x).q_\phi(y_2|x)$. The architecture of the model can easily be extended to the two high-level factor cases. This has been achieved by modifying the last layer of the C_ϕ network. Results obtained are shown in Fig.6.8 where the purpose is to transfer the glasses and the smile of $x_{y_1=1,y_2=1}$ to $x_{y_1=0,y_2=0}$.

6.5 Conclusion

Our CVQVAE approach clearly outperforms the state-of-the-art approaches, both in terms of disentanglement and in terms of generated image quality. Future works could adapt CVQVAE to the architecture of a hierarchical VQ-VAE (such as the one proposed in VQ-VAE2 [199]) and GAN (such as VQGAN[58]) so as to further improve the quality of generated images.

6.6 Supplementary Material

6.6.1 Sensibility analysis

As a reminder, the size of the latent variable z has been set in the article to $N_z \times N_z$ (with $N_z = 16$), and the embedding space is composed of K=512 vectors of dimension D=16. Finally, N_c has been set to 16. Tab. 6.4 illustrates the influence of the hyper-parameters N_z (first block), N_c (second block), K (third block) and D (last block) for the "glasses" label.

A small value of N_z ($N_z = 4$) reduces the quality of the generated images (FID increases) and hinders disentanglement properties (SR(+) decreases). Moreover, a high value of N_z increases the quality of the generated images (FID decreases) but also deteriorates the disentanglement properties of the model (SR(+) and SR(-) decrease). By increasing N_z (the capacity of z is increased), we run the risk that z encodes information that c should encode. Conversely, by reducing the size of z, we run the risk that z does not make it possible to encode all the useful information and that the model uses c to encode information that z should encode. We can thus intuit why N_z should neither be too small nor too large to obtain good disentanglement properties. Note that the influence of the size of the latent space (S_z) in the CGVAE2 approach (Tab. 6.5) is similar to the one observed for CVQVAE.

Finally, the influence of K, N_C and D is relatively weak as soon as they are chosen large enough.

Tab. 6.6 and 6.7 study the influence of β (see text after Eq. 6 in the article) for the CVQVAE (Tab. 6.6) and the CGVAE2 (Tab. 6.7) approaches. As a reminder, β weights the Kullback-Leibler divergence term. For the CVQVAE approach, it is interesting to note that the influence of β is relatively low, with respect to the quality of the generated images (please see FID column of Tab. 6.6 and Fig. 6.9). However, it is well-known that β may influence the disentanglement properties of the model. Our observation are consistent with the conclusion stated in β-VAE[86]: "when β is too low or too high, the model learns an entangled latent representation due to either too much or too little capacity in the latent z bottleneck." (increasing β may limit the capacity of z). Note that our values of β are very small according to the values of β mentioned in β-VAE: this is due to the fact that $\log(p_\theta(x|y,z,c))$ in Eq. 6 has been replaced (without loss of generality) by the mean squared error between x and $D_\theta(y,z,c)$. This is now what is generally done in most of the implementations of VAE.

Contrary to the CVQVAE case, β has a large influence on the quality of the generated images with CGVAE2: a too large value of β leads to images of low quality (please see the FID column of Tab. 6.7 and Fig. 6.10). However, in both cases, we can observe that β must not be too small nor too large in order to achieve good disentangle-

ment properties: for CGVAE2, best results are obtained with β = 1e-2. It is therefore necessary to balance the quality of the generated images (FID) and the disentanglement properties of the model (SR(-), SR(+)). $\beta = 1e-4$ seems to be a good choice. As shown in Fig. 6.10, results obtained with $\beta = 1e-2$ or with $\beta = 1e-3$ are blurred (the capacity of z is too small). Moreover, faces generated with $\beta = 1e-2$ or with $\beta = 1e-3$ are not as similar to the original face of $x_{y=0}$ as the face generated with $\beta = 1e-4$. Best results are clearly obtained for $\beta = 1e-4$.

Increasing β for CGVAE2 may produce disentanglement properties similar to CVQ-VAE, but at the expense of image quality. These results illustrate the relevance of using a discrete latent representation.

Table 6.4 Influence of the hyperparameters N_z, N_c, K, and D (for the "glasses" label) in the CVQVAE approach. For each block, the red color indicates the varying parameter, while results displayed in bold are those obtained with the parameter setting of the article.

N_z	N_c	K	D	SR(-)	SR(+)	CFD	FID
4×4	16	512	16	100%	65.11%	0.099	21.33
8×8	**16**	**512**	**16**	**100%**	**79.13%**	**0.093**	**20.05**
16×16	16	512	16	95.97%	69.96%	0.105	19.54
8×8	8	512	16	99.92%	71.36%	0.097	20.14
8×8	16	**512**	**16**	**100%**	**79.13%**	**0.093**	**20.05**
8×8	32	512	16	100%	79.35%	0.096	20.10
8×8	16	64	**16**	100%	67.97%	0.119	21.91
8×8	16	128	16	100%	72.01%	0.108	21.23
8×8	16	256	16	100%	77.84%	0.098	20.18
8×8	**16**	512	**16**	**100%**	**79.13%**	**0.093**	**20.05**
8×8	16	1024	16	99.92%	78.65%	0.098	20.23
8×8	16	512	2	96.68%	73.41%	0.117	21.34
8×8	16	512	4	98.05%	76.17%	0.103	20.99
8×8	16	512	8	99.85%	78.23%	0.099	20.44
8×8	**16**	**512**	16	**100%**	**79.13%**	**0.093**	**20.05**
8×8	16	512	32	100%	78.43%	0.099	20.04

6.6.2 Comparison with ELEGANT [265]

As shown in Fig.2.16, ELEGANT[265] is a GAN-based method that also allows to swap the face attributes of two images. Tab. 6.8 provides quantitative results that can be obtained with ELEGANT and CVQVAE for the "glasses" label and the "smile" one while Fig. 6.11 presents some swapping results. Since ELEGANT is fully supervised, the supervision rate has been set to 1 for CVQVAE in these experiment.

As evidenced by the very low value of LPIPS, ELEGANT yields reconstructed im-

Table 6.5 Influence of the size of the latent space (S_z) for the "glasses" label in the CGVAE2 approach. Results displayed in bold are those obtained with the parameter setting of the article.

S_z	SR(-)	SR(+)	CFD	FID
64	100%	62.74%	0.104	22.09
128	100%	66.19%	0.112	21.34
512	99.61%	70.36%	0.098	20.80
1024	**99.85%**	**72.83%**	**0.097**	**20.51**
2048	99.92%	64.42%	0.098	20.47

Table 6.6 Influence of β (for the "glasses" label) in the CVQVAE approach. Results displayed in bold are those obtained with the parameter setting of the article.

β	SR(-)	SR(+)	CFD	FID
$1e-6$	99.85%	69.03%	0.101	20.00
$1e-5$	99.92%	72.07%	0.098	20.24
1e − 4	**100%**	**79.13%**	**0.093**	**20.05**
$1e-3$	100%	77.34%	0.095	20.17

Table 6.7 Influence of β (for the "glasses" label) in the CGVAE2 approach. Results displayed in bold are those obtained with the parameter setting of the article. In order to observe the drop in performance for SR(+), we use larger β values than in CVQVAE.

β	SR(-)	SR(+)	CFD	FID
$1e-6$	44.96%	11.27%	0.137	20.29
$1e-5$	86.04%	64.43%	0.105	20.26
1e − 4	**99.85%**	**72.83%**	**0.097**	**20.51**
$1e-3$	100%	77.13%	0.101	22.34
$1e-2$	100%	78.97%	0.104	23.25
$1e-1$	100%	77.18%	0.074	57.72

Table 6.8 Results obtained with the "glasses" label (two first rows) and the "smile" one (two last rows). The criteria are the same than those used in Tab. 6.4

Model	SR(-)	SR(+)	LPIPS	FID
ELEGANT (glasses)	99.16%	95.66%	0.011	18.78
CVQVAE (glasses)	100%	80.26%	0.192	19.63
ELEGANT (smiling)	90.23%	91.25%	0.007	18.79
CVQVAE (smiling)	90.01%	90.33%	0.185	14.56

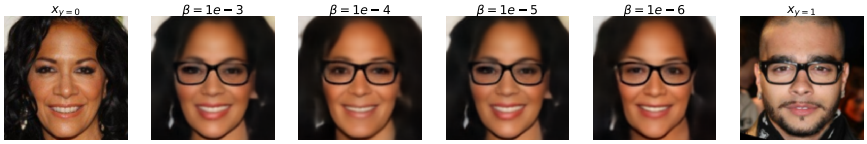


Fig. 6.9 Attribute swapping ("glasses label") using CVQVAE with different values of β. The second, third, fourth, and fifth images should be $x_{y=0}$ with the glasses of $x_{y=1}$. The value of β is given at the top of each result.

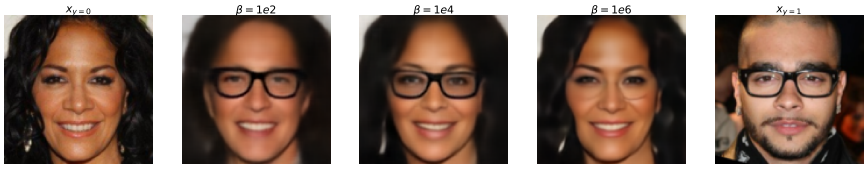


Fig. 6.10 Attribute swapping ("glasses label") using CGVAE2 with different values of β. The second, third, fourth, fifth, and sixth images should be $x_{y=0}$ with the glasses of $x_{y=1}$. The value of β is given at the top of each result. As a reminder, the generated images are obtained by feeding the decoder with z_0 (latent representation of $x_{y=0}$), c_1 (features of $x_{y=1}$), and $y = 1$. For $\beta = 1e - 1$, the capacity of z is so limited that c is actually used to model both the face and the glasses (the fake image is similar to $x_{y=1}$ even if z_0 is used for reconstruction). On the opposite, the capacity of z is too high for $\beta = 1e - 6$.

ages whose quality is superior to the proposed method: the reconstructed images are actually very similar to the original images. These good results may be explained by the fact that ELEGANT only learns the residual images. Moreover, it is based on a U-Net[206] structure whose skip connections enable to bring information about the original image to the decoder. This means that the original image is used at each step of the reconstruction: it is first used by the decoder to compute the residual image, and, then, the reconstruction is obtained by summing the residual image and the original image. On the opposite, our decoder takes only as input the latent variables, the label and its associated features.

The FID criterion that compares here the distribution of fake images (reconstructed images after attribute swapping) with the one of real images shows that the quality of the fake images is much less satisfactory for ELEGANT than the one of reconstructed images. Indeed, CVQVAE obtains a better FID than ELEGANT for the "smile" label. In the case of ELEGANT, we have observed that exchanging the features of two images often leads to reconstructions that are corrupted by artifacts, as illustrated in Fig. 6.11. We can note that making a face neutral or adding glasses often leads to images that are corrupted by strong artifacts. On the opposite, making a face smiling or removing glasses seems to be less prone to artifacts.

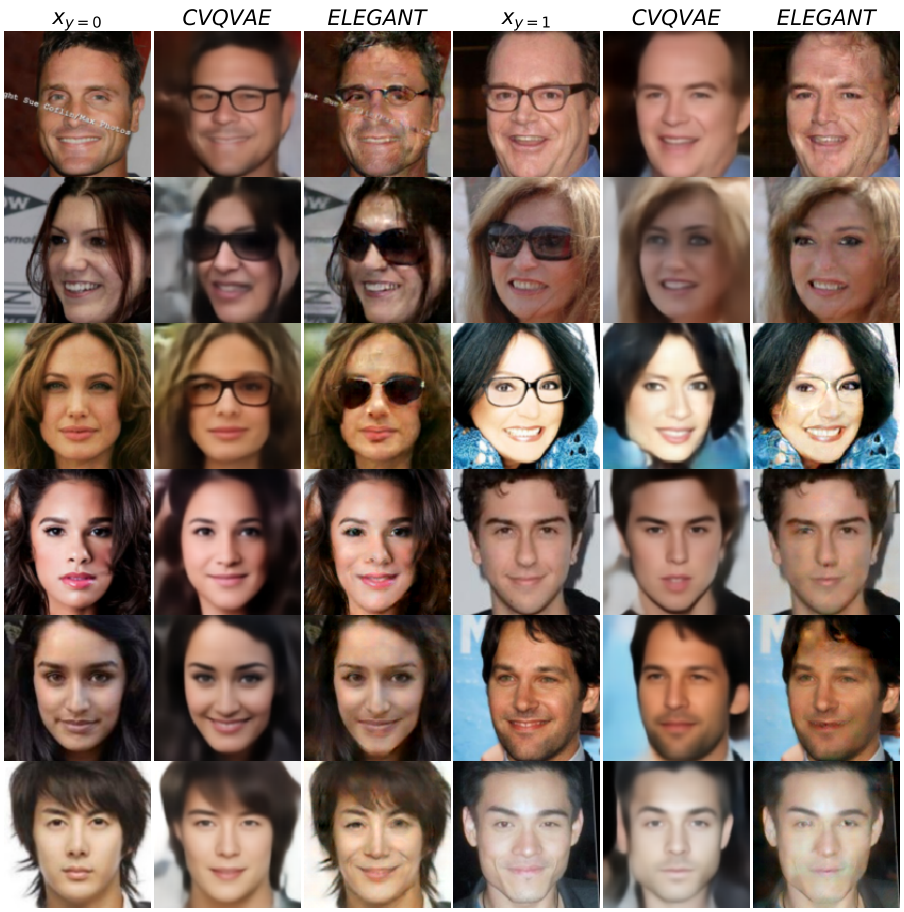


Fig. 6.11 Attribute swapping ("glasses" label for the three first rows and "smile label" for the three last rows) using CVQVAE and ELEGANT. For each row, the second and third images should be the face of $x_{y=0}$ with the glasses or the smile of $x_{y=1}$. The two rightmost images should be the face of $x_{y=1}$ without glasses or with the neutral attitude of $x_{y=0}$.

With respect to the success rates of swapping (SR(-) and SR(+)), ELEGANT and CVQVAE provide similar results, except for SR(+) in the case of the "glasses" label where ELEGANT outperforms the proposed method. However, the features are not very-well transferred from one image to another image in the case of ELEGANT. We can observe in Fig. 6.11 that CVQVAE transfers better the glasses of $x_{y=0}$ to $x_{y=1}$, or the smile of $x_{y=1}$ to $x_{y=0}$ or the neutral attitude of $x_{y=0}$ to $x_{y=1}$. This suggests that CVQVAE achieves a better disentanglement of the label and its features from the other variables.

To conclude, ELEGANT yields better reconstructions but is prone to artifacts (after attribute swapping). CVQVAE yields smoother images but its disentangling ability allows a better transfer of features between two images. There are other fundamental differences between these two methods. By learning the distribution of the faces, CVQVAE does not reconstruct rare or unseen features such as the watermark in the first row of Fig. 6.11. Such a property is of great interest in medical imaging where pathological

regions can be defined as those who are not well reconstructed. On the opposite, ELEGANT seems to be able to reconstruct these features. This is mainly because it learns the residual and it is based on a U-Net structure. Finally, CVQVAE can accomplish more tasks than ELEGANT. It provides a classifier. Moreover, it should be possible to generate new images by sampling directly from CVQVAE. After the training stage, this requires to learn three distributions: one for y, one for c given $y = 0$, and one for c given $y = 1$. Moreover, instead of sampling z from the uniform distribution in the VQVAE approach, note that [246] fits an autoregressive distribution (PixelCNN) over the values of z. Finally, note that ELEGANT is fully supervised whereas CVQVAE handles arbitrary supervision rates.

CHAPTER 7

Conclusions and Future work

In this concluding chapter, we provide a summary of the key findings and contributions of this research study, emphasizing their significance in both practical applications and methodological advancements. Additionally, we explore potential directions for future exploration within the field of generative models, specifically focusing on disentangled representation and conditional generation.

7.1 Summary and discussion

In this thesis, the main contributions can be summarised into two parts: the disentangled representation learning and conditional generation.

In Chapter 2, we extensively explored the utilization of generative models for disentangled representation learning and conditional generation. They can be applied in various application scenarios. And as the use of one of these models alone may be limited by its applicability to the scenario, more and more people are choosing to combine them. This combination of generative models has expanded the boundaries of their applications, as demonstrated by the discussions in Section 2.1.4. For instance, we delved into the concept of VAE-GAN [129] and diffuseVAE [176], highlighting how they harness the encoding capabilities of VAEs to enhance pure generative models such as GANs and diffusion models. Furthermore, Stable Diffusion [205], which has exhibited remarkable success in the domain of text-to-image generation, has been integrated with these three models. Initially, the VQGAN [58], comprising a combination of VQVAE [246] and GAN [69], was employed to achieve efficient compression capabilities. Subsequently, a diffusion model [89] was trained to generate the latent space conditioned on text inputs, effectively reducing computational resources. These novel combinations and applications of generative models exemplify the extent to which their potential is limited only by our imagination.

Chapter 3 presents a work related to the diffusion model, which represents one of the latest and most popular generative models in the field. The aim is to reveal the power and versatility of this model, particularly in the context of sequential data, despite

its primarily recognized application in 2D image generation. Through comprehensive validation, we establish that the diffusion model can be readily applied to sequential data, specifically for facial animation generation. This finding highlights its potential for broader applications on time series data. Moreover, we use a plug-and-play framework that capitalizes on the remarkable flexibility of the reverse process in the diffusion model. This framework entails training an unconditional diffusion model and subsequently conditioning the reverse process with various types of conditions. This leveraging of the diffusion model's reverse process enables us to incorporate different kinds of conditions including text, label, partial sequence, etc., enhancing the model's adaptability and expanding its range of applications.

Chapter 4 delves into the potential application of disentangled representation learning in the context of medical imaging. Specifically, we explore its applicability to medical images, focusing on the example of hip bones. Through a detailed analysis of the generated samples corresponding to each possible sex label of the hip bone, we highlight the distinct regions that differentiate between males and females. This approach allows for a targeted examination of the specific regions of interest associated with each label, shedding light on the possible anatomical differences that exist. To achieve this, we leverage the concept of (semi-)supervised VAE [122] that devises a methodology to separate the representation of sex information from identity information within the latent space. The architecture is shown in Fig.4.2. By modifying the label in the latent representation while preserving the identity information, it becomes possible to reconstruct the hip bone images of both sexes for the same individual.

We have also discussed in Chapter 4 that the selection of the appropriate model should be based on the specific application. In this task, we have chosen to utilize a VAE due to potential registration errors coming from the process of converting CT images into 3D meshes. The inherent capabilities of a VAE, such as its ability to remove high-frequency information and capture essential distributions, make it a suitable choice for generating the desired outputs.

The paradigm established in this chapter holds significant potential for broader application in the medical field, particularly when seeking to identify and highlight regions of interest related to specific diseases. By applying a similar approach to other diseases, it becomes possible to extract and visualize the distinctive regions associated with each condition. This offers valuable insights and can aid in both diagnosis and research by directing attention to the relevant areas that may exhibit significant variations or abnormalities.

However, the method presented in Chapter 4 only incorporates the label variable

into the latent representation. The label variable cannot fully capture the putative characteristics associated with the label. This limitation may not be apparent when dealing with the application presented in Chapter 4. However, in image domains, characteristics associated to the high-level factor, such as the type of glasses or the pathology of a disease can vary significantly. In Chapter 5, we propose a novel approach to address this issue by including the representation of high-level factors and the associated characteristics into the latent space. We introduce a new variable, denoted as u, which represents the features associated with these high-level factors within the VAE framework. We validate the effectiveness of our model by swapping the u values of different images and examining whether their corresponding features are successfully exchanged.

We enhance the work presented in Chapter 5 by introducing a discrete latent space and a two-step learning procedure in Chapter 6. This improvement leads to a significant enhancement in the quality and accuracy of the generated outputs, surpassing previous approaches such as CCVAE [111] and ELEGANT [265]. In Chapter 2, we present the method of CCVAE (illustrated in Fig.2.12), which can learn label-related representations but suffers from low quality and accuracy, restricting its practical applications. Similarly, ELEGANT (depicted in Fig.2.16) focuses solely on feature swapping, relying on a U-net architecture and adversarial training to enhance image generation quality. However, our experiments on ELEGANT reveal persistent artifacts in the generated images, consistent with observations from the original paper. In contrast, our method excels in accurately swapping features, even for rare characteristics, as demonstrated by the example of the first row in Fig.6.3 (a). Furthermore, we conduct further validation of the potential application of this method to medical imaging, as depicted in Fig.6.5. With the same idea as discussed in Chapter 4, we reconstruct a patient image as that of a normal person. By comparing the differences between the original and reconstructed images, we effectively highlight the pathology associated with the disease.

In conclusion, this thesis presents several generative models that contribute to disentangled representation learning and conditional generation. The proposed models offer promising solutions for various applications, including the field of medical imaging, specifically for interpretation and educational purposes. Through our experiments, we also explore the application of the state-of-the-art generative model, the diffusion model, and use a plug-and-play method for conditional generation. Overall, this research demonstrates the value and potential impact of generative models in advancing disentangled representation learning and enabling conditional generation for a wide range of applications, including the medical imaging domain.

7.2 Limitation and Future work

While our proposed generative models have shown promising results in disentangled representation learning and conditional generation, there are still some limitations that need to be addressed. In this section, we discuss these limitations and suggest potential avenues for future research and improvement.

Generation quality improvements For a long period, disentangled representation learning has primarily been applied to toy datasets. It always faces challenges when generating high-dimensional data and capturing high-frequency information. In our research presented in Chapter 5 and Chapter 6, we observed that these methods suffer from a loss of high-frequency information, which limit their application scenarios.

One potential future direction involves improving generation quality is to combine VAE with other generative models, such as GAN [129] or diffusion models [176]. This approach harnesses the encoding capability of VAE to learn disentangled representations, while integrating GAN or diffusion models to enhance the quality of generated outputs. By leveraging the strengths of each model, we can potentially achieve superior results in generation tasks.

Furthermore, the adoption of hierarchical generation has shown tremendous promise in elevating the quality of generated outputs, as demonstrated in [1, 113, 199]. By incorporating multiple levels of abstraction and progressively refining the generated samples, hierarchical generation can capture finer details and exhibit a more coherent and visually pleasing outcome. Consequently, this approach holds promise as a future avenue for advancing the quality of generation.

Finally, the requirement to achieve real disentangled representation may impose excessive constraints, resulting in the generation of low-quality images. Some researchers seek to explore the learning of latent transformations to achieve high-quality information generation [276, 96, 194], as opposed to relying on a shared latent space where high-level factors are controlled by specific latent dimensions. However, it is important to note that this approach typically requires a high-quality pretrained model to establish a robust latent space that can be effectively manipulated.

Learning the influence between different variables in disentangled representation

In the context of medical images, several factors including age, sex, and duration of illness play a significant role in influencing the observed pathology. While the disentanglement assumes independence of each latent factor, there are instances where these factors can interact with each other. For instance, in the case of Alzheimer's dis-

ease (AD), the severity of pathology tends to increase as an individual gets older. This implies a causal relationship between age and the pathology of AD.

To explore this, we can employ a Directed Acyclic Graph (DAG) to learn the relationships among the latent variables [272]. Once the relationships among these attributes are established, modifying one attribute will inevitably influence the others.

Possible application of diffusion model in medical image

Despite the considerable attention diffusion models have received in the field of image processing, their full potential has yet to be explored for medical images.

In Chapter 3, we have demonstrated the practicality of the diffusion model for addressing missing data. Specifically, it successfully predicts the remaining sequence based on partial input, making it a promising approach for handling missing data. Previous researches on medical image inpainting have primarily relied on GAN-based methods [11, 10], the emergence of the diffusion model has opened up new possibilities in this field. Recent work has applied the diffusion model to image inpainting tasks [154], but its potential for medical image inpainting remains largely unexplored.

Moreover, the versatility of the diffusion model extends beyond handling missing data in medical images. It can also be effectively employed in various other tasks, such as data augmentation and segmentation. In the context of data augmentation, the diffusion model can generate diverse synthetic data instances that can enrich the training dataset and improve the robustness of machine learning models. Additionally, in the domain of image segmentation, the diffusion model can be utilized to refine and enhance the accuracy of segmentation algorithms by generating high-quality segmentation masks. Thus, the diffusion model exhibits great potential in a wide range of applications

REFERENCES

[1] R. Abdal, Y. Qin, and P. Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[2] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 9525–9536, 2018.

[3] R. Agier, S. Valette, R. Kéchichian, L. Fanton, and R. Prost. Hubless keypoint-based 3D deformable groupwise registration. *Medical Image Analysis*, 59, 2020.

[4] Y. Alaluf, O. Tov, R. Mokady, R. Gal, and A. Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 18511–18521, 2022.

[5] B. AlBahar and J.-B. Huang. Guided image-to-image translation with bi-directional feature transformation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9016–9025, 2019.

[6] J. M. L. Alcaraz and N. Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *arXiv preprint arXiv:2208.09399*, 2022.

[7] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken elbo. In *International conference on machine learning*, pages 159–168. PMLR, 2018.

[8] A. Almahairi, S. Rajeshwar, A. Sordoni, P. Bachman, and A. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. In *International conference on machine learning*, pages 195–204. PMLR, 2018.

[9] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[10] K. Armanious, V. Kumar, S. Abdulatif, T. Hepp, S. Gatidis, and B. Yang. ipa-medgan: Inpainting of arbitrary regions in medical imaging. In *2020 IEEE international conference on image processing (ICIP)*, pages 3005–3009. IEEE, 2020.

[11] K. Armanious, Y. Mecky, S. Gatidis, and B. Yang. Adversarial inpainting of medical image modalities. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3267–3271. IEEE, 2019.

[12] N. T. Arun, N. Gaw, P. Singh, K. Chang, K. V. Hoebel, J. Patel, M. Gidwani, and J. Kalpathy-Cramer. Assessing the validity of saliency maps for abnormality localization in medical imaging. In *Medical Imaging with Deep Learning*, 2020.

[13] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[14] O. Avrahami, D. Lischinski, and O. Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022.

[15] V. Azizi, M. Usman, H. Zhou, P. Faloutsos, and M. Kapadia. Graph-based generative representation learning of semantically and behaviorally augmented floorplans. *The Visual Computer*, 38, 2022.

[16] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*, pages 2745–2754, 2017.

[17] D. Baranchuk, I. Rubachev, A. Voynov, V. Khrulkov, and A. Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

[18] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[19] D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[20] G. Bouritsas, S. Bokhnyak, S. Ploumpis, M. Bronstein, and S. Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7213–7222, 2019.

[21] G. Bouritsas, S. Bokhnyak, S. Ploumpis, M. Bronstein, and S. Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7213–7222, 2019.

[22] H. Bouzid and L. Ballihi. Facial expression video generation based-on spatio-temporal convolutional gan: Fev-gan. *Intelligent Systems with Applications*, page 200139, 2022.

[23] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[24] E. A. Brempong, S. Kornblith, T. Chen, N. Parmar, M. Minderer, and M. Norouzi. Denoising pretraining for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4175–4186, 2022.

[25] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[26] J. Brůžek, F. Santos, B. Dutailly, P. Murail, and E. Cunha. Validation and reliability of the sex estimation of the human os coxae using freely available DSP2 software for bioarchaeology and forensic anthropology. *American Journal of Physical Anthropology*, 164(2):440–449, 2017.

[27] A. Bulat, J. Yang, and G. Tzimiropoulos. To learn image super-resolution, use a gan to learn how to do image degradation first. In *Proceedings of the European conference on computer vision (ECCV)*, pages 185–200, 2018.

[28] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[29] D. Casas and M. A. Otaduy. Learning nonlinear soft-tissue dynamics for interactive avatars. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–15, 2018.

[30] R. T. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

[31] T. Chen, R. Zhang, and G. Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

[32] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[33] S. Cheng, I. Kotsia, M. Pantic, and S. Zafeiriou. 4dfab: A large scale 4d database for facial expression analysis and biometric applications. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5117–5126, 2018.

[34] Y.-C. Cheng, H.-Y. Lee, M. Sun, and M.-H. Yang. Controllable image synthesis via SegVAE. In *European conference on computer vision (ECCV)*, pages 159–174, 2020.

[35] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *SSST@EMNLP*, 2014.

[36] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.

[37] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[38] J. Chorowski, R. J. Weiss, S. Bengio, and A. Van Den Oord. Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM transactions on audio, speech, and language processing*, 27(12):2041–2053, 2019.

[39] A. Clark, J. Donahue, and K. Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

[40] D. Cosker, E. Krumhuber, and A. Hilton. A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In *2011 international conference on computer vision*, pages 2296–2303. IEEE, 2011.

[41] A. Creswell and A. A. Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018.

[42] Y. Deng, J. Yang, S. Xu, D. Chen, Y. Jia, and X. Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[43] E. L. Denton et al. Unsupervised learning of disentangled representations from video. *Advances in neural information processing systems*, 30, 2017.

[44] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.

[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[46] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[47] S. Dieleman, L. Sartran, A. Roshannai, N. Savinov, Y. Ganin, P. H. Richemond, A. Doucet, R. Strudel, C. Dyer, C. Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

[48] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[49] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[50] B. Dolhansky and C. C. Ferrer. Eye in-painting with exemplar generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7902–7911, 2018.

[51] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[52] X. Dong, Y. Yan, W. Ouyang, and Y. Yang. Style aggregated network for facial landmark detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–388, 2018.

[53] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[54] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[55] J. d'Oliveira Coelho and F. Curate. Cadoes: An interactive machine-learning approach for sex estimation with the pelvis. *Forensic Science International*, 302, 2019.

[56] F. Eitel and K. Ritter. Testing the robustness of attribution methods for convolutional neural networks in MRI-based Alzheimer's disease classification. In *Interpretability of Machine Intelligence in Medical Image Computing and Multimodal Learning for Clinical Decision Support*, Cham, 2019. Springer International Publishing.

[57] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. Technical Report 1341, University of Montreal, June 2009.

[58] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the conference on computer vision and pattern recognition (CVPR)*, pages 12873–12883, 2021.

[59] G. Fanelli, J. Gall, H. Romsdorfer, T. Weise, and L. Van Gool. A 3-d audio-visual corpus of affective communication. *IEEE Transactions on Multimedia*, 12(6):591–598, 2010.

[60] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 534–551, 2018.

[61] C. Ferrari, G. Lisanti, S. Berretti, and A. Del Bimbo. A dictionary learning-based 3d morphable shape model. *IEEE Transactions on Multimedia*, 19(12):2666–2679, 2017.

[62] R. Fong, M. Patrick, and A. Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *ICCV*, pages 2950–2958, 2019.

[63] Z. Gan, L. Li, C. Li, L. Wang, Z. Liu, J. Gao, et al. Vision-language pre-training: Basics, recent advances, and future trends. *Foundations and Trends® in Computer Graphics and Vision*, 14(3–4):163–352, 2022.

[64] L. Gao, D. Chen, Z. Zhao, J. Shao, and H. T. Shen. Lightweight dynamic conditional gan with pyramid attention for text-to-image synthesis. *Pattern Recognition*, 110:107384, 2021.

[65] R. Gao, Y. Song, B. Poole, Y. N. Wu, and D. P. Kingma. Learning energy-based models by diffusion recovery likelihood. *arXiv preprint arXiv:2012.08125*, 2020.

[66] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[67] S. Gershman and N. Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

[68] S. Gong, M. Li, J. Feng, Z. Wu, and L. Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

[69] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[70] A. Graikos, N. Malkin, N. Jojic, and D. Samaras. Diffusion models as plug-and-play priors. *arXiv preprint arXiv:2206.09012*, 2022.

[71] J. M. Graving and I. D. Couzin. Vae-sne: a deep generative model for simultaneous dimensionality reduction and clustering. *BioRxiv*, pages 2020–07, 2020.

[72] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022.

[73] J. Guan, C. Pan, S. Li, and D. Yu. Srdgan: learning the noise prior for super resolution with dual generative adversarial networks. *arXiv preprint arXiv:1903.11821*, 2019.

[74] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, and A. Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

[75] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[76] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proc. ACM Multimedia*, pages 2021–2029, 2020.

[77] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li. Towards fast, accurate and stable 3d dense face alignment. In *European Conference on Computer Vision*, pages 152–168. Springer, 2020.

[78] P. Guo, P. Wang, J. Zhou, V. M. Patel, and S. Jiang. Lesion mask-based simultaneous synthesis of anatomic and molecular mr images using a gan. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*, pages 104–113. Springer, 2020.

[79] X. Guo, J. W. Gichoya, S. Purkayastha, and I. Banerjee. Cvad: A generic medical anomaly detector based on cascade vae. *arXiv preprint arXiv:2110.15811*, 2021.

[80] C. Han, H. Hayashi, L. Rundo, R. Araki, W. Shimoda, S. Muramatsu, Y. Furukawa, G. Mauri, and H. Nakayama. Gan-based synthetic brain mr image generation. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 734–738. IEEE, 2018.

[81] W. Harvey, S. Naderiparizi, V. Masrani, C. Weilbach, and F. Wood. Flexible diffusion modeling of long videos. *arXiv preprint arXiv:2205.11495*, 2022.

[82] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[83] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019.

[84] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[85] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[86] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[87] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[88] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[89] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[90] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022.

[91] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[92] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

[93] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[94] E. Hoogeboom, V. G. Satorras, C. Vignac, and M. Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 8867–8887. PMLR, 2022.

[95] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang. Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks. *arXiv preprint arXiv:1704.00849*, 2017.

[96] X. Hu, Q. Huang, Z. Shi, S. Li, C. Gao, L. Sun, and Q. Li. Style transformer for image inversion and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11337–11346, 2022.

[97] C.-F. Huang and C.-Y. Huang. Emotion-based ai music generation system with cvae-gan. In *2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pages 220–222. IEEE, 2020.

[98] H. Huang, R. He, Z. Sun, T. Tan, et al. Introvae: Introspective variational autoencoders for photographic image synthesis. *Advances in neural information processing systems*, 31, 2018.

[99] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1501–1510, 2017.

[100] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018.

[101] A. Hyvärinen and P. Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[102] D. Im Im, S. Ahn, R. Memisevic, and Y. Bengio. Denoising criterion for variational auto-encoding framework. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[103] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[104] J. A. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. L. Ball, K. S. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. Langlotz, B. N. Patel, M. P. Lungren, and A. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. AAAI Conference on Artificial Intelligence, 2019.

[105] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[106] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[107] I. Jeon, W. Lee, M. Pyeon, and G. Kim. Ib-gan: Disentangled representation learning with information bottleneck generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7926–7934, 2021.

[108] Z.-H. Jiang, Q. Wu, K. Chen, and J. Zhang. Disentangled representation learning for 3d face shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11957–11966, 2019.

[109] B. Jing, G. Corso, J. Chang, R. Barzilay, and T. Jaakkola. Torsional diffusion for molecular conformer generation. *arXiv preprint arXiv:2206.01729*, 2022.

[110] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Confer-*

ence, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14, pages 694–711. Springer, 2016.

[111] T. Joy, S. Schmon, P. Torr, N. Siddharth, and T. Rainforth. Capturing label characteristics in VAEs. In *International Conference on Learning Representations, (ICLR)*, 2020.

[112] Y. Ju, J. Zhang, X. Mao, and J. Xu. Adaptive semantic attribute decoupling for precise face image editing. *The Visual Computer*, 37, 2021.

[113] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[114] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

[115] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[116] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[117] G. Kim, T. Kwon, and J. C. Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022.

[118] H. Kim and A. Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

[119] J. Kim, J. Kim, and S. Choi. Flame: Free-form language-based motion synthesis and editing. *arXiv preprint arXiv:2209.00349*, 2022.

[120] T. Kim and Y. Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.

[121] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International conference on machine learning*, pages 1857–1865. PMLR, 2017.

[122] D. Kingma, D. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, 2014.

[123] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[124] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[125] A. Kolesnikov, A. Dosovitskiy, D. Weissenborn, G. Heigold, J. Uszkoreit, L. Beyer, M. Minderer, M. Dehghani, N. Houlsby, S. Gelly, T. Unterthiner, and X. Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. In *Ninth International Conference on Learning Representations*, 2021.

[126] D. Komar and J. Buikstra. *Forensic Anthropology: Contemporary Theory And Practice*. Oxford University Press, New York, 2008.

[127] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

[128] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato. Fader networks: Manipulating images by sliding attributes. *Advances in neural information processing systems*, 30, 2017.

[129] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.

[130] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[131] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[132] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[133] B. Li, X. Qi, T. Lukasiewicz, and P. Torr. Controllable text-to-image generation. *Advances in Neural Information Processing Systems*, 32, 2019.

[134] B. Li, Y. Zhao, S. Zhelun, and L. Sheng. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1272–1279, 2022.

[135] H. Li, Y. Yang, M. Chang, S. Chen, H. Feng, Z. Xu, Q. Li, and Y. Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022.

[136] R. Li, S. Yang, D. A. Ross, and A. Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412, 2021.

[137] S. Li and Y. Sung. Inco-gan: variable-length music generation method based on inception model-based conditional gan. *Mathematics*, 9(4):387, 2021.

[138] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017.

[139] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022.

[140] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30, 2017.

[141] Y. Li, Z. Wang, L. Yin, Z. Zhu, G. Qi, and Y. Liu. X-net: a dual encoding–decoding method in medical image segmentation. *The Visual Computer*, 2021.

[142] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022.

[143] K. Lin, D. Li, X. He, Z. Zhang, and M.-T. Sun. Adversarial ranking for language generation. *Advances in neural information processing systems*, 30, 2017.

[144] Z. Lin, K. Thekumparampil, G. Fanti, and S. Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *international conference on machine learning*, pages 6127–6139. PMLR, 2020.

[145] H. Liu, Z. Wan, W. Huang, Y. Song, X. Han, and J. Liao. Pd-gan: Probabilistic diverse gan for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9371–9381, 2021.

[146] R. Liu, C. Subakan, A. H. Balwani, J. Whitesell, J. Harris, S. Koyejo, and E. L. Dyer. A generative modeling approach for interpreting population-level variability in brain structure. In *MICCAI*, pages 257–266, 2020.

[147] X. Liu, H. Huang, W. Wang, and J. Zhou. Multi-view 3d shape style transformation. *The Visual Computer*, 38, 2022.

[148] X. Liu, P. Sanchez, S. Thermos, A. O'Neil, and S. Tsaftaris. Learning disentangled representations in the imaging domain. *Medical Image Analysis*, 80, 06 2022.

[149] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[150] Z. Liu, P. Luo, X. Wang, and X. Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

[151] Z.-S. Liu, W.-C. Siu, and L.-W. Wang. Variational autoencoder for reference based image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 516–525, 2021.

[152] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124. PMLR, 2019.

[153] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[154] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[155] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11451–11461, 2022.

[156] S. Luo and W. Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021.

[157] Z. Lyu, Z. Kong, X. Xu, L. Pan, and D. Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *arXiv preprint arXiv:2112.03530*, 2021.

[158] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1445–1453, 2016.

[159] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[160] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[161] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[162] O. Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.

[163] T. C. Mok and A. C. Chung. Learning data augmentation for brain tumor segmentation with coarse-to-fine generative adversarial networks. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4*, pages 70–80. Springer, 2019.

[164] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4610–4617, 2019.

[165] P. Murail, J. Bruzek, F. Houët, and E. Cunha. DSP: A tool for probabilistic sex diagnosis using worldwide variability in hip-bone measurements. *Bulletins et mémoires de la Société d'Anthropologie de Paris*, 17(3-4):167–176, 2005.

[166] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine. Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31, 2018.

[167] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski. Plug and play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4467–4477, 2017.

[168] A. Nguyen, J. Yosinski, and J. Clune. *Understanding Neural Networks via Feature Visualization: A Survey*, pages 55–76. Springer International Publishing, Cham, 2019.

[169] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[170] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[171] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 16805–16827. PMLR, 2022.

[172] E. Nikita and P. Nikitas. Sex estimation: a comparison of techniques based on binary logistic, probit and cumulative probit regression, linear and quadratic discriminant analysis, neural networks, and naïve Bayes classification using ordinal variables. *International Journal of Legal Medicine*, 134(3):1213–1225, 2020.

[173] N. Nozawa, H. Shum, Q. Feng, E. S. L. Ho, and S. Morishima. 3d car shape reconstruction from a contour sketch using gan and lazy learning. *The Visual Computer*, 38, 2022.

[174] N. Otberdout, M. Daoudi, A. Kacem, L. Ballihi, and S. Berretti. Dynamic facial expression generation on hilbert hypersphere with conditional wasserstein generative adversarial nets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[175] N. Otberdout, C. Ferrari, M. Daoudi, S. Berretti, and A. Del Bimbo. Sparse to dense dynamic 3d facial expression generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20385–20394, 2022.

[176] K. Pandey, A. Mukherjee, P. Rai, and A. Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. *arXiv preprint arXiv:2201.00308*, 2022.

[177] T. Pang, C. Lu, C. Du, M. Lin, S. Yan, and Z. Deng. On calibrating diffusion probabilistic models. *arXiv preprint arXiv:2302.10688*, 2023.

[178] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021.

[179] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. Ieee, 2009.

[180] J. Peng, D. Liu, S. Xu, and H. Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784, 2021.

[181] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez. Invertible Conditional GANs for image editing. In *NIPS Workshop on Adversarial Training*, 2016.

[182] M. Petrovich, M. J. Black, and G. Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021.

[183] A. Phaphuangwittayakul, F. Ying, Y. Guo, L. Zhou, and N. Chakpitak. Few-shot image generation based on contrastive meta-learning generative adversarial network. *The Visual Computer*, 2022.

[184] M. Plappert, C. Mandery, and T. Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016.

[185] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[186] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, pages 8599–8608. PMLR, 2021.

[187] R. A. Potamias, J. Zheng, S. Ploumpis, G. Bouritsas, E. Ververas, and S. Zafeiriou. Learning to generate customized dynamic 3d facial expressions. In *Computer Vision – ECCV 2020: 16th European Conference*, page 278–294. Springer-Verlag, 2020.

[188] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10619–10629, 2022.

[189] A. R. Punnakkal, A. Chandrasekaran, N. Athanasiou, A. Quiros-Ramirez, and M. J. Black. Babel: Bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 722–731, 2021.

[190] C. Qi, X. Cun, Y. Zhang, C. Lei, X. Wang, Y. Shan, and Q. Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv preprint arXiv:2303.09535*, 2023.

[191] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[192] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[193] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[194] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[195] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[196] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European conference on computer vision (ECCV)*, pages 704–720, 2018.

[197] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European conference on computer vision (ECCV)*, pages 704–720, 2018.

[198] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, pages 725–741, 2018.

[199] A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, 2019.

[200] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016.

[201] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[202] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[203] M. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California, June 2016. Association for Computational Linguistics.

[204] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.

[205] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[206] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[207] A. Ruiz, O. Martinez, X. Binefa, and J. Verbeek. Learning disentangled representations with reference-based variational autoencoders, 2019.

[208] O. Rybkin, K. Daniilidis, and S. Levine. Simple and effective vae training with calibrated decoders. In *International Conference on Machine Learning*, pages 9179–9189. PMLR, 2021.

[209] O. Rybkin, K. Daniilidis, and S. Levine. Simple and effective VAE training with calibrated decoders, 2021.

[210] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[211] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[212] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[213] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[214] A. Segato, V. Corbetta, M. Di Marzo, L. Pozzi, and E. De Momi. Data augmentation of 3d brain environment using deep convolutional refined auto-encoding alpha gan. *IEEE Transactions on Medical Robotics and Bionics*, 3(1):269–272, 2020.

[215] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, 2019.

[216] H. Seo and G. Luo. Generating 3d facial expressions with recurrent neural networks. In *Intelligent Scene Modeling and Human-Computer Interaction*, pages 181–196. Springer International Publishing, 2021.

[217] W. Shen and R. Liu. Learning residual images for face attribute manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[218] R. Shetty, M. Rohrbach, L. Anne Hendricks, M. Fritz, and B. Schiele. Speaking the same language: Matching machine to human captions by adversarial training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4135–4144, 2017.

[219] N. Siddharth, B. Paige, J.-W. van de Meent, A. Desmaison, N. D. Goodman, P. Kohli, F. Wood, and P. H. S. Torr. Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[220] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.

[221] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[222] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[223] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. In *Workshop on Visualization for Deep Learning, ICML*, 2017.

[224] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[225] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[226] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[227] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. *Advances in neural information processing systems*, 29, 2016.

[228] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[229] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[230] Y. Song, L. Shen, L. Xing, and S. Ermon. Solving inverse problems in medical imaging with score-based generative models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[231] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[232] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[233] J. Tae, H. Kim, and T. Kim. Editts: Score-based editing for controllable text-to-speech. *arXiv preprint arXiv:2110.02584*, 2021.

[234] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin. Variational auto-encoder-based detection of electricity stealth cyber-attacks in ami networks. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1590–1594. IEEE, 2021.

[235] M. Tao, H. Tang, F. Wu, X.-Y. Jing, B.-K. Bao, and C. Xu. Df-gan: A simple and effective baseline for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16515–16525, 2022.

[236] G. Tevet, B. Gordon, A. Hertz, A. H. Bermano, and D. Cohen-Or. Motionclip: Exposing human motion generation to clip space. *arXiv preprint arXiv:2203.08063*, 2022.

[237] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

[238] H. Tingfei, C. Guangquan, and H. Kuihua. Using variational auto encoding in credit card fraud detection. *IEEE Access*, 8:149841–149853, 2020.

[239] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[240] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017.

[241] X. Tu, J. Zhao, M. Xie, Z. Jiang, A. Balamurugan, Y. Luo, Y. Zhao, L. He, Z. Ma, and J. Feng. 3d face reconstruction from a single image assisted by 2d face images in the wild. *IEEE Transactions on Multimedia*, 23:1160–1172, 2020.

[242] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2018.

[243] H. Uzunova, J. Ehrhardt, and H. Handels. Generation of annotated brain tumor mris with tumor-induced tissue deformations for training and assessment of neural networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23*, pages 501–511. Springer, 2020.

[244] A. Vahdat and J. Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020.

[245] D. Valevski, M. Kalman, Y. Matias, and Y. Leviathan. Unitune: Text-driven image editing by fine tuning an image generation model on a single image. *arXiv preprint arXiv:2210.09477*, 2022.

[246] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[247] S. Van Steenkiste, F. Locatello, J. Schmidhuber, and O. Bachem. Are disentangled representations helpful for abstract visual reasoning? *Advances in Neural Information Processing Systems*, 32, 2019.

[248] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[249] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29, 2016.

[250] B. Wallace, A. Gokul, and N. Naik. Edict: Exact diffusion inversion via coupled transformations. *arXiv preprint arXiv:2211.12446*, 2022.

[251] C. Wang, H. Zheng, Z. Yu, Z. Zheng, Z. Gu, and B. Zheng. Discriminative region proposal adversarial networks for high-quality image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 770–785, 2018.

[252] L. Wang, V. Sindagi, and V. Patel. High-quality facial photo-sketch synthesis using multi-adversarial networks. In *2018 13th IEEE international conference on automatic face and gesture recognition (FG 2018)*, pages 83–90. IEEE, 2018.

[253] Q. Wang, T. Artières, M. Chen, and L. Denoyer. Adversarial learning for modeling human motion. *The Visual Computer*, 36, 2020.

[254] S. Wang, Y. Zou, W. Min, J. Wu, and X. Xiong. Multi-view face generation via unpaired images. *The Visual Computer*, 38, 2022.

[255] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[256] W. Wang, X. Alameda-Pineda, D. Xu, P. Fua, E. Ricci, and N. Sebe. Every smile is unique: Landmark-guided diverse smile generation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 7083–7092, 2018.

[257] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[258] Y. Wang, P. Bilinski, F. Bremond, and A. Dantcheva. G3an: Disentangling appearance and motion for video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

[259] Y. Wang, P. Bilinski, F. Bremond, and A. Dantcheva. Imaginator: Conditional spatio-temporal gan for video generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1160–1169, 2020.

[260] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou. Diffusion-gan: Training gans with diffusion. *arXiv preprint arXiv:2206.02262*, 2022.

[261] J. Wen, H. Ma, and X. Luo. Deep generative smoke simulator: connecting simulated and real data. *The Visual Computer*, 36, 2020.

[262] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[263] J. Wu, Z. Huang, D. Acharya, W. Li, J. Thoma, D. P. Paudel, and L. V. Gool. Sliced wasserstein generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3713–3722, 2019.

[264] T. Xiao, J. Hong, and J. Ma. DNA-GAN: Learning disentangled representations from multi-attribute images. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2018.

[265] T. Xiao, J. Hong, and J. Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 168–184, 2018.

[266] J. Xie, Y. Lu, S.-C. Zhu, and Y. Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644. PMLR, 2016.

[267] J. Xu, X. Wang, W. Cheng, Y.-P. Cao, Y. Shan, X. Qie, and S. Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. *arXiv preprint arXiv:2212.14704*, 2022.

[268] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[269] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang. Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation. *Neuroinformatics*, 16:383–392, 2018.

[270] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *European conference on computer vision (ECCV)*, pages 776–791, 2016.

[271] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *arXiv preprint arXiv:2207.09983*, 2022.

[272] M. Yang, F. Liu, Z. Chen, X. Shen, J. Hao, and J. Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9593–9602, 2021.

[273] R. Yang, P. Srivastava, and S. Mandt. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022.

[274] T. Yang, Y. Wang, Y. Lv, and N. Zh. Disdiff: Unsupervised disentanglement of diffusion probabilistic models. *arXiv preprint arXiv:2301.13721*, 2023.

[275] X. Yang, Y. Li, H. Qi, and S. Lyu. Exposing gan-synthesized faces using landmark locations. In *Proceedings of the ACM workshop on information hiding and multimedia security*, pages 113–118, 2019.

[276] X. Yao, A. Newson, Y. Gousseau, and P. Hellier. A latent transformer for disentangled face editing in images and videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13789–13798, 2021.

[277] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation, 2017.

[278] Z. Yin, K. Xia, S. Wang, Z. He, J. Zhang, and B. Zu. Unpaired low-dose ct denoising via an improved cycle-consistent adversarial network with attention ensemble. *The Visual Computer*, 2022.

[279] T. Yoshikawa, Y. Endo, and Y. Kanamori. Diversifying detail and appearance in sketch-based face image synthesis. *The Visual Computer*, 38, 2022.

[280] K. Young, G. Booth, B. Simpson, R. Dutton, and S. Shrapnel. Deep neural network or dermatologist? *Lecture Notes in Computer Science*, 2019.

[281] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[282] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11304–11314, 2022.

[283] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[284] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018.

[285] L. Zhang and M. Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.

[286] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022.

[287] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[288] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, and P. Liu. A high-resolution spontaneous 3d dynamic facial expression database. In *IEEE workshops on automatic face and gesture recognition*, pages 1–6, 2013.

[289] Y. Zhang, C. Ong, J. Zheng, L. S.-T., and G. Z. Generative design of decorative architectural parts. *The Visual Computer*, 38, 2022.

[290] Z. Zhang and L. Schomaker. Dtgan: Dual attention generative adversarial networks for text-to-image generation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[291] Q. Zhao, E. Adeli, N. Honnorat, T. Leng, and K. M. Pohl. Variational autoencoder for regression: Application to brain aging analysis. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 823–831. Springer International Publishing, 2019.

[292] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[293] L. Zhou, Y. Du, and J. Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021.

[294] J. Zhu, D. Zhao, B. Zhang, and B. Zhou. Disentangled inference for gans with latently invertible autoencoder. *International Journal of Computer Vision*, 130(5):1259–1276, 2022.

[295] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 597–613. Springer, 2016.

[296] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[297] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. *Advances in neural information processing systems*, 30, 2017.

[298] M. Zhu, P. Pan, W. Chen, and Y. Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5802–5810, 2019.

[299] X. Zhu, L. Zhang, L. Zhang, X. Liu, Y. Shen, and S. Zhao. Gan-based image super-resolution with a novel quality loss. *Mathematical Problems in Engineering*, 2020:1–12, 2020.

[300] K. Zou, S. Faisan, F. Heitz, , and S. Valette. Joint disentanglement of labels and their features with VAE. In *IEEE International Conference on Image Processing (ICIP)*, 2022.

[301] K. Zou, S. Faisan, F. Heitz, M. Epain, P. Croisille, L. Fanton, and S. Valette. Disentangled representations: towards interpretation of sex determination from hip bone. *The Visual Computer*, 2023.

[302] K. Zou, S. Faisan, F. Heitz, and S. Valette. Disentangling high-level factors and their features with conditional vector quantized vaes. *Pattern Recognition Letters*, 2023.

[303] K. Zou, S. Faisan, B. Yu, S. Valette, and H. Seo. 4d facial expression diffusion model. *arXiv preprint arXiv:2303.16611*, 2023.

LIST OF PUBLICATIONS

1. K. Zou, S. Faisan, F. Heitz, , and S. Valette. Joint disentanglement of labels and their features with VAE. In IEEE International Conference on Image Processing (ICIP), 2022.

2. K. Zou, S. Faisan, F. Heitz, M. Epain, P. Croisille, L. Fanton, and S. Valette. Disentangled representations: towards interpretation of sex determination from hip bone. The Visual Computer, 2023.

3. K. Zou, S. Faisan, F. Heitz, and S. Valette. Disentangling high-level factors and their features with conditional vector quantized vaes. Pattern Recognition Letters, 2023

4. K. Zou, S. Faisan, B. Yu, S. Valette, and H. Seo. 4d facial expression diffusion model. arXiv preprint arXiv:2303.16611, 2023. 162

Appendices

Appendix A

Why maximinzing the ELBO is equivalent to minimizing $D_{KL}(q_\phi(x,z)||p_\theta(x,z))$

To ensure that the approximated latent distribution aligns with the prior defined on the latent space and that the generated data resembles the observed data, the objective function of the VAE can be formulated by considering the KL divergence between these two joint distributions:

$$\begin{aligned} D_{KL}(q_\phi(x,z)\|p_\theta(x,z)) &= \iint q_\phi(x,z)\log\frac{q_\theta(x,z)}{p_\theta(x,z)}dzdx \\ &= \int q_D(x)\left[\int q_\phi(z\mid x)\log\frac{q_D(x)q_\phi(z\mid x)}{p_\theta(x,z)}dz\right]dx, \\ &= \mathbb{E}_{x\sim q_D(x)}\left[\int q_\phi(z\mid x)\log\frac{q_D(x)q_\phi(z\mid x)}{p_\theta(x,z)}dz\right] \end{aligned} \tag{A.1}$$

where q_D is defined in Eq.2.7. The objective is to minimize the KL divergence between the two distributions, aiming to make them as similar as possible. It is important to note that both $q_\phi(x,z)$ and $p_\theta(x,z)$ have parameters (θ and ϕ) that can be learned. By optimizing these parameters, the distributions are trained to minimize the KL divergence and achieve a high degree of similarity.

We can easily observe that $\log\frac{q_D(x)q_\phi(z|x)}{p_\theta(x,z)} = \log q_D(x) + \log\frac{q_\phi(z|x)}{p_\theta(x,z)}$, and we have:

$$\begin{aligned} \mathbb{E}_{x\sim q_D(x)}\left[\int q_\phi(z\mid x)\log q_D(x)dz\right] &= \mathbb{E}_{x\sim q_D(x)}\left[\log q_D(x)\int q_\phi(z\mid x)dz\right] \\ &= \mathbb{E}_{x\sim q_D(x)}[\log q_D(x)] = C \end{aligned} \tag{A.2}$$

C stands for a constant, so this term can be ignored. Then Eq. (A.1) can be simplified as:

$$
\begin{aligned}
\mathcal{L} &= \mathbb{E}_{x\sim q_D(x)}\left[\int q_\phi(z \mid x)\log\frac{q_\phi(z \mid x)}{p_\theta(x \mid z)p(z)}dz\right] \\
&= \mathbb{E}_{x\sim q_D(x)}\left[-\int q_\phi(z \mid x)\log p_\theta(x \mid z)dz + \int q_\phi(z \mid x)\log\frac{q_\phi(z \mid x)}{p(z)}dz\right] \\
&= \mathbb{E}_{x\sim q_D(x)}\left[\mathbb{E}_{z\sim q_\phi(z|x)}[-\log p_\theta(x \mid z)] + D_{KL}(q_\phi(z \mid x)\|p(z))\right]
\end{aligned}
\tag{A.3}
$$

Hence, we arrive at the ultimate objective of the VAE. This is essentially the negative of the ELBO. The distinction between Equation (A.3) and Equation (2.5) lies in the optimization direction: we aim to maximize the ELBO, whereas in this case, we seek to minimize the KL divergence.

Appendix B

Hyperparameter Setting for DDPM

Based on Eq. 2.14, we can write $p(x_t|x_{t-1})$ with reparameterization trick we have:

$$\begin{aligned} x_t &= \sqrt{\alpha_t} x_{t-1} + \sqrt{\beta_t}\epsilon_t \\ &= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{\beta_{t-1}}\epsilon_{t-1}\right) + \sqrt{\beta_t}\epsilon_t \\ &= \cdots \\ &= \left(\sqrt{\alpha_t}\cdots\sqrt{\alpha_1}\right) x_0 + \underbrace{\left(\sqrt{\alpha_t}\cdots\sqrt{\alpha_2}\right)\sqrt{\beta_1}\epsilon_1 + \cdots + \sqrt{\alpha_t}\sqrt{\beta_{t-1}}\epsilon_{t-1} + \sqrt{\beta_t}\epsilon_t}_{\text{the sum of multiple Gaussian noise}} \end{aligned} \tag{B.1}$$

The mean of the sum of multiple Gaussian noise is 0, the variances is $\left(\alpha_t \cdots \alpha_2\right)\beta_1 + \left(\alpha_t \cdots \alpha_3\right)\beta_2 + \cdots + \alpha_t\beta_{t-1} + \beta_t$. With $\alpha_t + \beta_t = 1$ (for all t),the sum of coefficients in each term of Eq. (B.1) becomes 1 (the variance then can be expressed as $1 - \sqrt{\alpha_t}\cdots\sqrt{\alpha_1}$):

$$\left(\alpha_t \cdots \alpha_1\right) + \left(\alpha_t \cdots \alpha_2\right)\beta_1 + \left(\alpha_t \cdots \alpha_3\right)\beta_2 + \cdots + \alpha_t\beta_{t-1} + \beta_t = 1 \tag{B.2}$$

As a result, we can rewrite x_t as:

$$x_t = \underbrace{\left(\sqrt{\alpha_t}\cdots\sqrt{\alpha_1}\right)}_{\text{note as } \sqrt{\bar{\alpha}_t}} x_0 + \underbrace{\sqrt{1 - \left(\sqrt{\alpha_t}\cdots\alpha_1\right)^2}}_{\text{note as } \sqrt{\bar{\beta}_t}}\bar{\epsilon}_t, \quad \bar{\epsilon}_t \sim \mathcal{N}(0, I) \tag{B.3}$$

We can observe that setting $\alpha_t + \beta_t = 1$ greatly facilitates the computation of x_t and also reduces the number of hyperparameters by half.

We now need to determine the value of α_t. Our goal is to minimize the KL divergence between two joint distributions given in Eq. (2.17). Ideally, we would like p and q to be equal, which would mean that their marginal distributions are also equal:

$$\begin{aligned} p(x_T) &= \int q\left(x_T \mid x_{T-1}\right)\cdots q\left(x_1 \mid x_0\right)\tilde{q}\left(x_0\right) dx_0 dx_1 \cdots dx_{T-1} \\ &= \int q\left(x_T \mid x_0\right)\tilde{q}\left(x_0\right) dx_0 \end{aligned} \tag{B.4}$$

It's important to note that $\tilde{q}(x_0)$ is the distribution of the real data and can be any arbitrary distribution. Therefore, in order for the above equation to remain true, we can only set $q(x_T|x_0) = p(x_T)$, which is independent of x_0 and follows a standard normal distribution ($p(x_T) \sim \mathcal{N}(0, I)$). Since we need to satisfy $q(x_T|x_0) \sim \mathcal{N}(0, I)$, we must select the appropriate value for α_t to ensure that $\bar{\alpha_T}$ is approximately zero. In the case of DDPM, this is done by setting $\alpha_t = \sqrt{1 - \frac{0.02t}{T}}$.

Regarding σ_t, different optimal values may correspond to different data distributions $\tilde{q}(x_0)$. To illustrate this point, let's consider two simple examples. First, suppose that the training set only contains one sample $\hat{x}$, which means that $\tilde{q}(x_0)$ is the Dirac distribution $\delta(x - \hat{x})$. In this case, the optimal value for σ_t^2 can be calculated as $\frac{\bar{\beta}t-1}{\bar{\beta}t}\beta_t$. Second, suppose that the data distribution $\tilde{q}(x_0)$ follows the standard normal distribution. In this scenario, the optimal value for σ_t^2 simplifies to just β_t. [89] claims that both of these settings for σ yield similar results. In Chapter 3, we set σ_t^2 equal to β_t

Appendix C

The objective of diffusion model

We can rewrite the objective function at arbitrary step t in Eq. 2.18 as:

$$\begin{aligned} &-\int q\left(x_T \mid x_{T-1}\right) \cdots q\left(x_1 \mid x_0\right) \tilde{q}\left(x_0\right) \log p_\theta\left(x_{t-1} \mid x_t\right) dx_0 dx_1 \cdots dx_T \\ =&-\int q\left(x_t \mid x_{t-1}\right) q\left(x_{t-1} \mid x_0\right) \tilde{q}\left(x_0\right) \log p_\theta\left(x_{t-1} \mid x_t\right) dx_0 dx_{t-1} dx_t. \end{aligned} \tag{C.1}$$

For $q\left(x_t \mid x_{t-1}\right)$, we have:

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{\beta_t} \varepsilon_t. \tag{C.2}$$

For $q\left(x_{t-1} \mid x_0\right)$, we have

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{\bar{\beta}_{t-1}} \varepsilon_{t-1}. \tag{C.3}$$

The contribution of the term $\log p_\theta\left(x_{t-1} \mid x_t\right)$ is as follows:

$$\frac{1}{2\sigma_t^2} ||x_{t-1} - \mu_\theta(x_t)||. \tag{C.4}$$

Based on the expression $q\left(x_t \mid x_{t-1}\right)$, we have $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \beta_t \epsilon_t$, then we can naturally set:

$$\mu_\theta(x_t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \epsilon'_\theta(x_t, t)). \tag{C.5}$$

With all the aforementioned equations, the objective can be written as:

$$\frac{\beta_t}{\alpha_t \sigma_t^2} \mathbb{E}_{\varepsilon_{t-1}, \varepsilon_t \sim \mathcal{N}(0,I), x_0 \sim \tilde{q}(x_0)} \left[\left\| \varepsilon_t - \epsilon'_\theta \left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{\alpha_t \bar{\beta}_{t-1}} \varepsilon_{t-1} + \sqrt{\beta_t} \varepsilon_t, t \right) \right\|^2 \right] \tag{C.6}$$

As the number of random variables to be sampled increases, it becomes more challenging to accurately estimate the loss function, which can be equivalently stated as the increasing volatility (variance) of the loss function estimates at each sampling instance.

We can observe that $\sqrt{\alpha_t\bar{\beta}_{t-1}}\bar{\varepsilon}_{t-1} + \sqrt{\beta_t}\varepsilon_t$ is equivalent to $\sqrt{\bar{\beta}_t}\varepsilon \mid \varepsilon \sim \mathcal{N}(0, I)$. Similarly, we can state that $\sqrt{\beta_{t-1}}\bar{\varepsilon}_{t-1} - \sqrt{\alpha_t\bar{\beta}_{t-1}}\varepsilon_t$ is equivalent to $\sqrt{\bar{\beta}_t}\omega \mid \omega \sim \mathcal{N}(0, I)$.

$$\varepsilon_t = \frac{\left(\sqrt{\beta_t}\varepsilon - \sqrt{\alpha_t\bar{\beta}_{t-1}}\omega\right)\sqrt{\bar{\beta}_t}}{\beta_t + \alpha_t\bar{\beta}_{t-1}} = \frac{\sqrt{\beta_t}\varepsilon - \sqrt{\alpha_t\bar{\beta}_{t-1}}\omega}{\sqrt{\bar{\beta}_t}} \tag{C.7}$$

Then we can write Eq. C.6 by replacing the variable ε_t and ε_{t-1} with ε and ω. It writes:

$$\begin{aligned}
&\frac{\beta_t}{\alpha_t\sigma_t^2}\mathbb{E}_{\omega,\varepsilon\sim\mathcal{N}(0,I)}\left[\left\|\frac{\sqrt{\beta_t}\varepsilon - \sqrt{\alpha_t\bar{\beta}_{t-1}}\omega}{\sqrt{\bar{\beta}_t}} - \epsilon'_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{\bar{\beta}_t}\varepsilon, t\right)\right\|^2\right] \\
=&\frac{\beta_t^2}{\bar{\beta}_t\alpha_t\sigma_t^2}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I),x_0\sim\tilde{q}(x_0)}\left[\left\|\varepsilon - \sqrt{\frac{\bar{\beta}_t}{\beta_t}}\epsilon'_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{\bar{\beta}_t}\varepsilon, t\right)\right\|^2\right] + C
\end{aligned} \tag{C.8}$$

This is the final objective of diffusion models. To simplified it, as discussed in Chapter 2, we can set x_{t-1} as follows:

$$\mu_\theta(x_t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, t)). \tag{C.9}$$

By removing the coefficient before Eq. C.8 and the constant C, we can derive a simplified loss function:

$$\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I),x_0\sim\tilde{q}(x_0)}\left[\left\|\varepsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{\bar{\beta}_t}\varepsilon, t\right)\right\|^2\right] \tag{C.10}$$

Kaifeng ZOU

Advancements in Generative Models: Enhancing Interpretability and Control of Complex Data through Disentanglement and Conditional Generation

Résumé

Les modèles génératifs sont une classe de modèles d'apprentissage automatique qui visent à apprendre la distribution sous-jacente d'un ensemble de données donné et à générer de nouveaux points de données qui ressemblent aux données originales. Ces modèles ont suscité beaucoup d'attention ces dernières années en raison de leur capacité à produire des échantillons de données réalistes et diversifiés. Les modèles génératifs, tels que les VAE (Variational Autoencoders), les GANs (Generative Adversarial Networks), les EBMs (Energy-Based Models), les modèles de diffusion, ont montré un grand potentiel dans de nombreux domaines, notamment la génération d'images, la synthèse de la parole et le traitement du langage naturel, et continuent d'être un domaine actif de recherche, avec de nouveaux modèles et techniques en développement pour améliorer leurs performances et élargir leurs applications. Une des applications les plus importantes des modèles génératifs est la représentation désentrelacée, qui fait référence à un type d'apprentissage des caractéristiques dans lequel les facteurs sous-jacents ou les attributs des données sont appris et représentés de manière indépendante. Dans notre recherche, nous utilisons des représentations désentrelacées pour relever le défi de la détermination du sexe et fournir des informations sur les résultats de classification. Cela est réalisé en générant des os de hanche pour le même individu des deux sexes, puis en effectuant une comparaison pour identifier les distinctions liées au sexe. De plus, nous visons à acquérir des connaissances sur le facteur de haut niveau et ses attributs en apprenant la représentation associée, ce qui nous permet de contrôler efficacement les caractéristiques liées à l'étiquette. Pour ce faire, nous introduisons deux cadres VAE innovants visant à apprendre la représentation associée à l'étiquette et à améliorer simultanément la qualité de la génération VAE. De plus, notre recherche contribue également à la génération conditionnelle. Nous appliquons un modèle de diffusion aux données séquentielles, montrant sa capacité à générer des expressions faciales 3D, impliquant des données en série temporelle. Ce processus inversé offre une flexibilité remarquable, permettant divers types de conditionnement et de génération grâce à une seule procédure de formation.

Résumé en anglais

Generative models are a class of machine learning models that aim to learn the underlying distribution of a given dataset and generate new data points that resemble the original data. These models have gained significant attention in recent years due to their ability to produce realistic and diverse samples of data. Generative models, such as VAEs (Variational Autoencoders) , GANs (Generative Adversarial Networks), EBMs (Energy-Based Models), diffusion models, have shown significant promise in many fields, including image generation, speech synthesis, and natural language processing, and continue to be an active area of research, with new models and techniques being developed to improve their performance and broaden their applications. One of the most important application of generative model is disentangled representation, which refers to a type of feature learning in which the underlying factors or attributes of data are learned and represented independently. In our research, we utilize disentangled representations to tackle the challenge of sex determination and provide insights into the classification results. This is achieved by generating hip bones for the same individual from both sexes and subsequently conducting a comparison to identify sex-related distinctions. Additionally, we aim to acquire knowledge about the high-level factor and its attributes by learning the associated representation, allowing us to effectively control label-related characteristics. To achieve this, we introduce two innovative VAE frameworks aimed at learning the label-associated representation and enhancing VAE's generation quality simultaneously. Additionally, our research also makes a contribution to conditional generation. We apply a diffusion model to sequential data, showcasing its ability to generate 3D facial expressions, which involve time series data. This reverse process provides remarkable flexibility, enabling various types of conditioning and generation through a single training procedure.