Multimodal Conditioning for Controllable Image and Video Generation

Soon Yau Cheong

Principal Supervisor: Dr. Andrew Gilbert
Co-supervisor: Dr. Armin Mustafa

This dissertation is submitted for the degree of
Doctor of Philosophy

Faculty of Arts, Business and Social Sciences, Music and Media
University of Surrey

Februrary 2025

I would like to dedicate this thesis to the memory of my parents
who sadly passed away during my PhD journey.
Their love and unwavering belief in me continue to inspire and motivate me every day.

To my wife Siew, thank you for your endless patience,
love, and support, especially during the most challenging moments.

To my sons Zenpo and Zenson, you remind me every day of the importance of perseverance and wonder.

Abstract

The field of generative AI has progressed at a rapid pace and can now produce high-quality images and videos from text prompts. This evolution has also led to greater user demand for precise control over the outcomes, posing new challenges in effectively directing generation processes. Standard conditioning techniques, mainly using text and image inputs, have proven useful but remain limited in handling more complex requirements, such as specific human pose, camera orientation or fine-grained visual appearance. This PhD research enhances conditioning techniques by introducing a **parametric approach** that emphasises **multimodal conditioning** for **image and video generation** models. It focuses on developing methods to enable more comprehensive user control, incorporating various modalities such as pose and spatial inputs to improve alignment between model capabilities and user intentions across different aspects of generation. By refining the conditioning mechanisms, this research aims to bridge the gap between user specifications and model outputs, ensuring greater flexibility, precision, and coherence in generated content.

This thesis presents several key contributions. Traditional methods for human pose conditioning, which rely on skeleton images, contain substantial redundancy and are computationally inefficient for modern architectures. To address this, we proposed the concept of **pose token**, where raw pose parameters are compressed into tokens that can be used as conditioning elements via attention mechanisms, a common approach in advanced architectures. We validated this token-based approach with both 2D body keypoints and 3D body parameters, demonstrating its effectiveness across multiple architectures, from transformers to diffusion models. Additionally, our parametric approach introduces groundbreaking techniques for human and camera pose interpolation within image generation.

A common approach for conditioning diffusion models involves incorporating adapters - lightweight models to deliver control signals to pre-trained image models. However, our research has revealed that this method often introduces a critical issue of **mode conflict**. This problem, worsened by cascading multiple adapters, results from an imbalance in control signals: the model can become dominated by one adapter, limiting the generative power of both the base model and other adapters. Despite its prevalence, this issue remains largely unaddressed in existing research. To solve this, we devised a **unified adapter architecture** that integrates both structural and visual conditioning within a single, harmonised control pathway. This unified approach delivers balanced multimodal conditioning, avoiding the pitfalls of adapter cascade and enabling greater model flexibility. As a result, our approach's high controllability empowers versatile human image generation and editing tasks.

Our research in 2D image generation was extended to video generation. Our study demonstrated that the architectural differences in transformer-based diffusion models make existing camera control methods for U-Net-based diffusion models ineffective. Through extensive experimentation, optimal architectures and camera representations were identified. Combined with our novel **camera motion guidance**, camera control was restored for **video diffusion transformers**, with motion boosted by over 400%. Our research on human pose conditioning for images extends to video generation. Unlike existing methods that require

detailed camera pose input for every frame, our approach achieves smooth video motion with minimal input. By specifying only the initial and final camera poses, our system interpolates between frames to produce continuous camera movements, enabling consistent, controlled video generation with reduced data requirements. This sparse video conditioning approach significantly simplifies the user interaction while ensuring fluid transitions and stable pose dynamics across frames, pushing the boundaries of efficient and user-friendly video generation.

Many of the challenges we aimed to address were novel, often lacking established evaluation methods. As a result, we proposed **new evaluation metrics** to rigorously assess these areas. One of these, People Count Error (PCE), identifies a unique type of error specific to AI-generated human images, such as inaccurate body part generation. This metric has already gained traction in the research community and is being adopted in image generation benchmarks, helping to set new standards for evaluating AI-driven human image quality.

Contents

Acknowledgements

I would like to express my deepest gratitude to my principal supervisor, Dr. Andrew Gilbert, for his invaluable guidance and support throughout my PhD. His expertise and advice helped shape the direction of this thesis, and I am especially thankful for his efforts in setting up the GPUs, which enabled me to run the essential experiments for this research. I am also grateful to my secondary supervisor, Dr. Armin Mustafa, for her thorough revisions and feedback on my conference papers, which greatly improved their quality and impact. A special thanks to Dr. Chun-Hao Huang, my internship supervisor at Adobe, for his constant encouragement and contributions to the code, which were instrumental to the success of the video generation project. I also extend my appreciation to Dr. Dyugu Ceylan at Adobe for her collaboration and insightful discussions during the project.

Declaration

I hereby declare that this thesis, along with the research and findings presented herein, is the result of my own efforts and original work. Any ideas, data, images, or text derived from the work of others—whether published or unpublished—are fully acknowledged and cited within the text, bibliography, or in footnotes, ensuring appropriate attribution to the respective originators. This thesis has not been submitted, in whole or in part, for any other academic degree or professional qualification.

Soon Yau Cheong

8 Februrary 2025

Chapter 1

Introduction

Image generation has long been a holy grail in computer vision, with applications spanning creative industries like advertising, gaming, and film production, as well as medical imaging and design. This pursuit drives research aimed at enhancing the quality and realism of generated images. A breakthrough occurred with the introduction of Generative Adversarial Networks (GANs) [31], which revolutionised the field by enabling the generation of high-quality images through adversarial training techniques. However, these early GANs primarily operated in an unconditional manner, generating images from random noise without any user-guided input. Even when some control was introduced, it was usually confined to a single input modality, such as generating images based solely on class labels [76] or predefined categories. This lack of control limited their practical applications, as users had little to no influence over the content, style, or specific characteristics of the generated images. Consequently, these initial models struggled to meet the demands of tasks requiring tailored output, rendering them less useful in real-world scenarios.

To address these limitations, subsequent advancements introduced image-to-image translation, a paradigm where GANs could generate images based on an input image. This innovation allowed for more directed generation, making it possible to convert sketches into photorealistic images or transfer artistic styles from one image to another. However, the reliance on single visual input meant that these models often lacked the flexibility to specify other important aspects. For example, while sketch-to-human models like [117] can generate photorealistic representations of humans that accurately reflect the desired pose based on the input image, the resulting appearances are random and lack controllability. On the other hand, the need for paired image-to-image data significantly limits the size and diversity of training data, ultimately limiting the GAN model's ability to generalise across different tasks and produce high-quality results in broader applications.

The introduction of text-to-image (T2I) transformers [120] DALL-E [90] at the start of this PhD study marked a significant leap forward in generative capabilities. Training on large text-image-pair datasets enables the creation of more expressive models by providing rich, diverse data that allows models to learn complex relationships between textual descriptions and visual content. This enhanced the model's ability to generate detailed, contextually accurate images that align closely with user-provided text. However, despite these improvements, challenges remained in controlling certain aspects of the output. For example, fine-grained details such as clothing textures, intricate human poses, and specific camera angles can be difficult to articulate through text alone, leading to results that may not fully capture the user's intent or desired aesthetic.

Soon after, the emergence of diffusion models [40], replaced autoregressive transformers as the preferred architecture for image generation due to their superior ability to generate high-quality images through iterative refinement. More recently, transformers have begun replacing U-Nets [96] as the denoising backbone in diffusion models due to their scalability. However, this architectural shift introduces new challenges in conditioning techniques. For example, camera control methods designed for U-Net-based diffusion models [35, 127] have proven ineffective for transformer-based diffusion models [5], highlighting the need for novel conditioning strategies tailored to these architectures.

1.1 Problem Statements

The several key challenges in effectively controlling generative models, which are addressed in this work, can be outlined in the following problem statements:

- **Limited Modality for Precise Control**: Many existing models rely on text or image as input modalities, which have their inherent limitations and are often not suitable for precise control of complex attributes such as angle of a camera, the articulation of a human pose, or fine-grained object details.
- **Challenges in Simultaneous Multimodal Conditioning**: Many existing models are restricted to a single input modality, which constrains the control over generated content. However, naively applying multiple conditions simultaneously can create conflicts among them, leading to mode conflict, where one or more controls become ineffective. For instance, in a model that incorporates both text and pose inputs, the pose control may overshadow the text control, resulting in an image with an accurately rendered human pose but failing to capture the desired human appearance or background as specified by the text prompt.
- **Rapidly Evolving Model Architectures Demand New Conditioning Methods**: As generative models evolve from GANs to more sophisticated architectures like transformers and diffusion models, adapting conditioning methods to these varied frameworks requires renewed effort. Each architecture handles inputs and conditioning differently, necessitating the development of tailored techniques to effectively integrate and control multimodal inputs. What works for one model type may not translate well to others due to the architectural difference.

1.2 Motivation and Objectives

The motivation behind this research stems from the limitations of current generative models in achieving precise control over generated content. As generative architectures evolve, there is a growing need for more sophisticated control mechanisms that can integrate various input modalities. By focusing on enhancing control over human and camera poses alongside traditional text and image conditioning, we aim to empower users to generate richer and more contextually accurate images and videos. This research seeks to bridge the gap between user intent and generated outcomes, ultimately advancing the capabilities of generative AI.

To achieve these goals, the following specific objectives have been identified:

1. **Exploration of New Control Modalities:** To enhance user control over generated content, we aim to investigate new control modalities that incorporate pose control for human and camera, focusing on innovative representations rather than relying on traditional 2D images.

2. **Harmonious Multimodal Conditioning:** Another key objective is to develop conditioning methods that enable the simultaneous and harmonious conditioning of poses alongside text and image inputs. This approach will allow for a more integrated and intuitive user experience, where users can specify poses in conjunction with descriptive text or reference images. By achieving this level of integration, we aim to facilitate more coherent and contextually rich image generation.

3. **Conditioning Across Various Model Architectures:** Our research will also focus on applying these conditioning methods to a range of modern model architectures - transformers and diffusion models. This objective aims to ensure that the developed control mechanisms are versatile and adaptable, allowing for effective integration across different generative frameworks. By examining how these methods perform within various architectures, we can refine our approach and contribute to the broader field of generative AI.

4. **Camera Pose Control in Video Generation:** Finally, we aim to apply camera pose control to video generation, focusing on transformer-based diffusion models (DiT) [85], which are rapidly gaining popularity over traditional U-Net-based diffusion models. This objective involves developing techniques that enable users to manipulate camera rotation angles and translation in generated videos with the new model architecture.

1.3 Contributions

Throughout the course of this research, several key contributions were made to address the challenges and limitations in multimodal image and video generation. These contributions span new evaluation metrics, advancements in pose control, improvements in multimodal conditioning across different model architectures, and solutions to issues such as mode conflict and camera control inefficiencies. Below is an overview of the main contributions:

1. **Parametric Pose Control**. Instead of the traditional method of controlling human poses with 2D images, we proposed parametric pose control approaches using pose tokens - a method compatible with modern architecture utilising attention mechanism. The idea was validaed with 2D body keypoint [16] before extending to 3D body SMPL parameters in [17], demonstrating the first simultaneous body and camera poses interpolation.

2. **Harmonious Multimodal Conditioning**. We were the first to propose multimodal conditioning of text and pose inputs in image generation [16] using transformer architectures. We extended these modalities by incorporating image inputs in [17, 18] for diffusion model architectures, enabling harmonious multimodal conditioning of text, visual, and pose inputs. This approach achieves unparalleled versatility and control in the generative process.

3. **Solving Mode Conflict**. We discovered a widespread mode conflict issue affecting existing text-and-visual prompt image generation models that use multiple adapter branch architectures

[53, 77, 136, 143]. This issue often causes the model to fixate on one controlling modality, rendering other modality controls ineffective. To address this, we proposed a unified single adapter branch architecture [18] for effective mitigation.

4. **Solving Ineffectiveness of Camera Control in Video Diffusion Transformers**. We demonstrated that existing camera control methods developed for U-net-based diffusion models [35, 127] are ineffective when applied to transformer-based diffusion models [82], resulting in limited or uncontrolled motion in the generated videos. To address this, we devised an effective method, Camera Motion Guidance (CMG), which successfully restores controllability and enhances motion generation by over 400%.

5. **Proposed a Novel Evaluation Metric**: A new evaluation metric, *People Count Error (PCE)* [16] was proposed, a method designed to detect artifacts unique to AI-generated human images. This metric has been widely adopted in the performance benchmarking of human image generation models [53, 65, 121, 137], contributing to more rigorous and standardised evaluations within the field.

1.4 List of Publications

This thesis incorporates findings from my conference and workshop manuscripts, which include:

- *Chapter 3: Parametric Human Pose Token for Autoregressive Transformer*

  Soon Yau Cheong, Armin Mustafa, and Andrew Gilbert. Kpe: Keypoint pose encoding for transformer-based image generation. In British Machine Vision Conference (BMVC), 2022 [16].

- *Chapter 4: Fine-grained Visual and 3D Pose Control for Diffusion Models*

  Soon Yau Cheong, Armin Mustafa, and Andrew Gilbert. Upgpt: Universal diffusion model for person image generation, editing and pose transfer. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 4173–4182, 2023 [17] .

- *Chapter 5: Avoiding Mode Conflict with Unified Pose-Visual Diffusion Adapter*

  Soon Yau Cheong, Armin Mustafa, and Andrew Gilbert. Visconet: Bridging and harmonizing visual and textual conditioning for controlnet. In ECCV Workshop Proceedings, 2024 [18].

- *Chapter 6: Guiding Camera Motion in Video Diffusion Transformer*

  Soon Yau Cheong, Dugyu Ceylan, Armin Mustafa, Andrew Gilbert, and Chun-Hao Paul Huang. Boosting Camera Motion Control for Video Diffusion Transformers. Arxiv Preprint 2410.10802. [15].

Chapter 2

Background

In this chapter, we provide an overview of two major generative model methods used in this thesis - transformers, and diffusion models—each utilising distinct conditioning techniques. We will explore how these conditioning methods are applied within their respective architectures, using common input modalities—text and images—as illustrative examples from existing research. Before delving into these architectures, we briefly review earlier generative models. These early approaches laid important theoretical foundations for modern generative techniques, influencing advancements in controllability and model expressiveness.

Next, we review the conditioning methods employed in prominent models, starting with direct feature fusion as used in conditional GANs. We then introduce the transformer architecture and attention mechanism to demonstrate how a text-to-text model can be adapted for text-to-image generation. Finally, we explore diffusion models, which have emerged as the leading architecture for image generation, and discuss how to control conditioning strength through diffuser guidance. Finally, we introduce common human and camera pose representations, along with relevant datasets used in this thesis.

2.1 Early Generative Models

Before the rise of transformers[120], and diffusion models[40], several early generative models provided key insights into probabilistic modelling and deep learning-based synthesis.

Restricted Boltzmann Machines (RBMs)[38] that use a bipartite architecture with visible and hidden units to model binary-valued data. RBMs were historically significant in pretraining deep neural networks and inspired later advancements in unsupervised learning. However, due to training challenges and the emergence of more scalable architectures, RBMs have largely been replaced by deep generative models.

Energy-Based Models (EBMs)[115] formulate generative modelling as an energy minimisation problem, where a neural network assigns an energy score to each data sample. Lower-energy configurations correspond to more likely samples, and learning involves reducing the energy of real samples while increasing the energy of unrealistic ones. While EBMs offer flexibility in modelling complex distributions, they typically require computationally expensive sampling methods such as Markov Chain Monte Carlo (MCMC) for training.

Variational Autoencoders (VAEs)[58] introduced probabilistic latent variable models by combining deep learning with Bayesian inference. VAEs consist of an encoder network that maps input data to a probabilistic latent space and a decoder that reconstructs the original data from sampled latent representations. The training objective maximises the Evidence Lower Bound (ELBO), which balances reconstruction accuracy and the regularisation of the latent space. Despite their advantages in structured latent representations, VAEs tend to produce blurry images due to their reliance on Gaussian latent spaces and the nature of the reconstruction loss.

Normalising flows [93] addressed some of the limitations of VAEs by learning an invertible transformation between a simple prior distribution (e.g., Gaussian) and complex data distributions. These models employ a sequence of bijective transformations parameterised by neural networks, allowing exact likelihood estimation and efficient sampling. Popular variants like Glow [57] use flow-based transformations for high-resolution image generation. However, normalising flows often require deep architectures to model complex distributions, making them computationally expensive.

Autoregressive models, such as PixelCNN[118], define a probability distribution over data by factorising it into a sequential product of conditional distributions. PixelCNN models images as a raster-scan sequence of pixels, predicting each pixel value based on previously generated ones. This explicit likelihood-based approach ensures stable training and high-quality samples but small receptive field of convolutional kernels limit the learning of long-range pixel relationship.

Introduced in 2014, Generative Adversarial Networks (GANs) [31] has been fundamental in the development of generative models. GANs consist of two neural networks - generator G and discriminator D. Starting from a low dimensional latent vector (random noise) z, the generator reshape and upsample the features with multiple convolutional layers [87] to create a *fake* image $G(z)$. The discriminator is a binary classifier made up of downsampling convolutional neural network, determining if a image is *real* image x or a *fake* image created by the generator. In training, the objective is to maximise the generator's ability to convince the discriminator that the generated images are real while minimising the likelihood of the discriminator misclassifying them using the adversarial loss [31]. Despite their success, GANs suffer from instability during training, mode collapse, and sensitivity to hyperparameters, limiting their robustness and generalisation capabilities.

These early approaches introduced essential ideas in probabilistic modelling, latent representations, and network training dynamics, influencing the development of subsequent architectures in modern generative modelling.

2.2 Direct Feature Fusion through the Lens of GANs

Direct feature fusion is the process of combining multiple features into a single representation. It can be used to condition neural networks by injecting conditioning inputs into the model's internal embeddings, typically through addition or concatenation with the conditioning embeddings. This technique is broadly applicable across deep neural network architectures and has been widely explored in GANs to control the generation process. Conditioning methods for GANs have continued to evolve and are now integrated into modern models, including transformers and diffusion models.

In early work, conditional GAN (cGAN) [76] conditions the image generation on class labels such as number digits. The one-hot-encoded label vector is concatenated with the latent noise vector at the model's input. In addition to conditioning at the input, [151] also explored injecting the conditioning signal into the model's internal embeddings through concatenation, demonstrating early success.

However, the basic method limits conditioning to a categorical nature *e.g.* number digits, object class labels and facial attributes. In a seminal paper, Pix2pix [48] replaces the traditional noise input vector with an image as the conditioning input, thereby formulating GANs for image-to-image translation tasks. The input image serves as a versatile and general form of structural conditioning, providing essential spatial and compositional guidance for the generative model. This type of conditioning can also take several image forms, including sketches, human skeleton pose, or segmentation maps, each offering different levels of abstraction and structural detail. These image-based inputs guide the generation process by defining the layout, pose, or structure of the scene or objects within it, allowing for precise control over the output while leaving room for the model to add necessary details like texture, colour, and finer attributes.

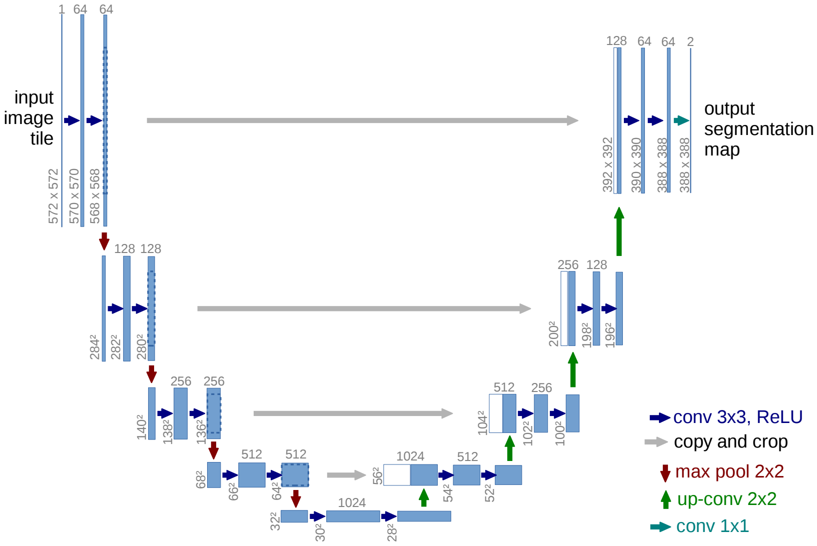


Figure 2.1: The block diagram of GAN architecture, reprinted from [96].

To handle the dimensional shift from a 1D latent vector z to 1D or image inputs $x \in \mathbb{R}^{H\times W\times C}$, the authors adopt the U-Net [96] architecture for the generator, which is characterised by its U-shaped structure, as shown in Figure 2.1. U-Net consists of a downsampling path that captures image context at different resolutions, followed by a symmetric expanding path through upsampling. The skip connections between the downsampling and upsampling paths help preserve spatial information that may be lost during downsampling, allowing the network to retain fine-grained details from the input image. U-Net has also been adopted as the backbone denoising model in early diffusion models due to its ability to capture multiscale features.

However, the authors of [48] found that conditioning random noise alongside the input image was ineffective, as the random noise was largely ignored by the model, leading to reduced randomness and less varied image outputs. To address this, [152] proposed injecting the spatially replicated latent noise into the intermediate layers of the generator's encoder, rather than at the model input. The injected latents were fused by concatenating them with the intermediate layer features along the channel dimension. Although initially used to inject noise rather than conditioning, this feature fusion technique of injecting latents into intermediate layers laid the groundwork for more advanced conditioning methods in later architectures.

There have also been efforts to inject conditioning into the normalisation layers of GANs, most notably for controlling image artistic styles [25, 46, 52, 54]. While these methods were influential, particularly in tasks involving artistic style manipulation, they are less prevalent in the newer model architectures used in our studies. As a result, they are not explored in detail in this thesis.

Overall, the input and internal embedding concatenation forms the foundation of conditioning in neural networks. However, concatenation increases the feature dimensions, altering the model layer dimensions and complicating the reuse of pre-trained model weights. As a result, linear layers have become a common solution to re-project the fused embeddings back to their original dimensions, allowing the preservation of the pre-trained architecture while incorporating a control mechanism.

2.3 Text Conditioning and Processing

Transformers have become the dominant architecture for text processing, offering powerful capabilities for capturing contextual relationships. This makes them the foundation of modern text-to-image models, enabling more accurate and flexible image generation. Before exploring the details of transformers, however, we briefly examine the history of text-to-image generation using GANs.

2.3.1 Text-to-Image GANs

There were early attempts to use GANs to perform text-to-image generation. StackGAN [139] adopts the aforementioned conditioning methods by concatenating text embedding at the generator input and also the the intermediate layers at the discriminator. AttnGAN [131] also explored using attention models [4] to retrieve the relevant word vectors for generating different sub-regions of the image. The earlier works in GANs used a small text encoder that trains on small text dataset, hence struggled to capture the complexity of text descriptions, often resulting in images that did not align well with the provided text prompts.

However, despite leveraging pre-trained large transformers as text encoders in models like XMC-GAN[138] and StyleGAN-T [55], GANs continues to lag behind diffusion models in text-to-image tasks [55]. Overall, GANs have had limited success in this domain, particularly when compared to more advanced architectures such as autoregressive transformers and diffusion models. These newer approaches, which this thesis focuses on, provide enhanced control, scalability, and image richness, effectively addressing the limitations of GANs in handling complex multimodal inputs like text descriptions

2.3.2 Transformers

The transformer architecture, first introduced in the groundbreaking paper “Attention is All You need” [120] revolutionised the field of natural language processing (NLP) and quickly became the de facto architecture for LLM [10, 21, 88] and language vision models (LVMs) [62, 86]. The utilisation of attention mechanisms to handle long-range dependencies of texts, leading to significant improvements in tasks like machine translation, text generation, and question-answering. The transformer architecture is highly scalable and their computation can be perform in parallel, enabling more efficient, larger and more powerful language models. The attention mechanism, which will be described later, particularly cross-attention has been adopted as a effective conditioning method for multimodal models.

2.3.3 Text Tokenisation

This section begins by providing an overview of text pre-processing, a critical step in preparing raw text for use in machine learning models. Pre-processing involves cleaning and normalising the text, converting it into a format that can be understood by a neural network. This typically includes operations such as removing special characters, lowercasing, and, most importantly, tokenisation.

Tokenisation breaks down text into smaller units, such as words or subword units, depending on the chosen tokenisation method. A popular tokenisation technique is Byte Pair Encoding (BPE) [103], which balances between word-level and character-level tokenisation by splitting rare words into subword units while keeping common words intact. BPE works by iteratively merging the most frequent pair of bytes (characters or character sequences) in a corpus, creating new tokens that represent these subword units. This technique helps to reduce the size of the vocabulary while still maintaining a good representation of uncommon words or word forms. An token ID is then assigned to each subword in the dictionary. The tokenisation process is illustrated in Figure 2.2. The purpose of masking will be described later in the section.

```
Text = "transformer is slower than CNN"
Subwords = ['transform', 'er', 'is', 'slow', 'er', 'than', 'c', 'n', 'n']
Tokens = [40, 6, 85, 2, 6, 13, 5, 35, 35]
Masked text = "transformer is <MASK> <MASK> CNN "
Masked tokens = [40, 6, 85, <MASK>, <MASK>, <MASK>, 5, 35, 35]
```

Figure 2.2: Illustration of tokenisation and masking in LLM training.

Once the text is tokenised, the tokens are converted into text embedding. Embeddings are dense tensor representations that capture the semantic meaning of words or tokens in a continuous vector space. These embeddings, matching the dimension of the model, are then fed into the model, enabling it to process and understand the textual information in a structured way, crucial for both natural language understanding and generation tasks. To emphasize their discrete nature, throughout this thesis, we refer to text embeddings and image embeddings as text tokens and image tokens, respectively. This distinction underscores the unique characteristics of these representations in the context of multimodal conditioning and generative tasks, facilitating a clearer understanding of how these tokens interact within the model architecture.

2.3.4 Transformer Architecture

A transformer is typically composed of multiple stacked blocks, each containing key components: attention layers, normalisation layers, and feedforward layers as shown in Figure 2.3a. At the inputs, positional encoding is added to the input tokens to inject information about the order of the tokens to help the model understand the position of each token within sequence. A common positional encoding

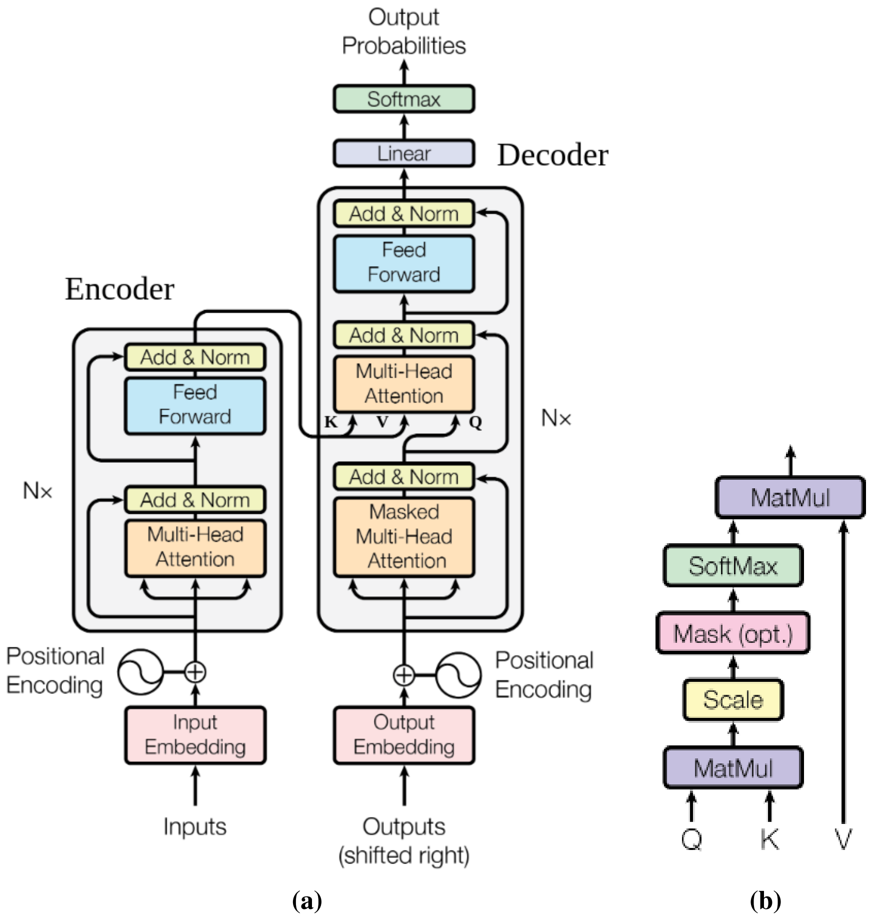


Figure 2.3: Transformer (a) architecture, (b) attention layer, adapted from [120].

is sinusoidal positional encoding, which can be described by the following equations:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{2.1}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{2.2}$$

where pos is the position of the token, i is the dimension index, d_{model} is the model's embedding dimension.

The embeddings are then fed into multiple blocks consisting of multi-head attention, normalisation and feedforward layer. The attention layer is the core mechanism in transformers, dynamically weighting relationships between tokens based on their relevance to one another. It enables the model to focus on different parts of an input sequence and capture dependencies, regardless of distance within the sequence. In the multi-head attention setup, the model divides the token embeddings into multiple subspaces, computes attention scores in each, and then combines the outputs. For example, if $d_{model} = 512$ and $head = 8$, then each head operates in a subspace of 64 dimensions. The equation for attention is defined as below and illustrated in Figure 2.3b:

$$M = Map(Q, K) = softmax(\frac{QK^T}{\sqrt{d_{model}}}) \tag{2.3}$$

$$Attention(M, V) = M \cdot V \tag{2.4}$$

where Q, K, and V are the query, key, and value projected from the input tokens. It starts by computing the attention map (Equation 2.3) to produce similarity score that indicates how relevant each key (and thus its corresponding value) is to the query. The result is a matrix of size $n \times n$ (where n is the number of tokens in the input sequence) representing how much attention each word should pay to every other word in the sequence. The scaling of $\frac{1}{\sqrt{d_{model}}}$ helps to prevent the dot product from becoming too large when the dimensionality is high, which could result in extremely small gradients during training. The softmax function is then applied to normalize these scores into a probability distribution, ensuring that they sum to 1. This normalised matrix, or attention map, reflects the relevance of each token to the others, giving larger weights to more contextually important tokens.

After computing the attention map, the next step is to multiply it by the V matrix (Equation 2.4). The V matrix contains the actual information that the model will use to update the output. The attention map essentially acts as a set of learned weights that determine how much of the information in the value matrix should be passed on for each token in the sequence. Each head performs this calculation, and the results are concatenated and linearly transformed back into the original dimension. When the queries Q, K and V are all derived from the same set of input tokens, it is referred to as *self-attention*.

The original transformer architecture consists of an encoder and a decoder. The encoder's output embeddings are fed into the decoder's multi-head attention's K and V while the Q comes from the decoder's input. This is known as *cross attention*, allowing the decoder to focus on specific parts of the input tokens. The encoder-decoder architecture can be used for machine translation *e.g.* English to French. All the English input tokens are ordered in sequence and fed into the encoder, while the decoder start with a special token $< SOS >$ (start of sentence) token. After a forward pass, from the model's output probabilities, the most likely text token is selected and appended to the outputs (decoder's input) after SOS. The newly formed sequence, now including the predicted token, is fed back into the decoder to generate the subsequent token. This *autoregressive* approach continues until an end-of-sequence token $< EOS >$ is produced.

Transformers can be trained in various ways, depending on the task at hand and the model's objective. The two most prominent types of transformer training are autoregressive training and masked language modeling (MLM), both of which are fundamental to popular LLM models. Each training strategy has distinct goals and is suited for different downstream tasks.In autoregressive training, the model learns to predict the next token in a sequence given the previous tokens. This approach is used by models like GPT [10, 88]. The model is trained to generate text in a sequential manner, where each token is predicted based on the tokens that came before it.

BERT's[21] training is based on MLM. During training, a certain percentage of tokens in the input sequence are randomly replaced with a special <MASK> token. The model is then tasked with predicting the original tokens based on the surrounding context, effectively learning relationships and dependencies within the text. This is illustrated in Figure 2.2. After pre-training, this learned representation can be fine-tuned for downstream task such as Question and Answer (Q&A). Instead of randomly masking words in the sequence, the question words are kept intact, while only the answer words are masked. This way, the model is specifically trained to predict the 'masked' answers based on the unmasked question and surrounding context. By doing this, the training focuses on teaching the model to understand and identify the relationship between the question and the answer within the passage, all made possible with the attention mechanism.

Transformers has a remarkable property that makes it the dominant neural network architecture - its scalability. Since each transformer block has the same input and output dimension, the model size and hence capacity can be increased by simply increasing the number of stacked blocks. This flexibility enables transformers to accommodate larger datasets and more complex tasks, improving performance without requiring fundamental architectural changes. Notably, a key difference in network topology between transformers and U-Net lies in the feature dimensions; in transformers, the feature dimensions remain consistent after each block, whereas in U-Net, they change throughout the network via downsampling and upsampling paths. This difference can significantly impact the strength and effectiveness of the conditioning methods, as will be discussed in Chapter 6.

2.3.5 Image Tokenisation

Building on its success in LLMs, the transformer architecture has been adapted for vision tasks. This repurposing for text-to-image generation signifies a major advancement in generative AI, facilitating the development of multimodal models encompassing visual and textual information.

2.3.5.1 Continuous Image Tokens

Inspired by text tokenisation, vision transformer (ViT) [24] introduced the concept of image tokenisation, where an image is divided into fixed-sized patches that are then flattened into one-dimensional vectors and projected into a lower-dimensional space. By using image tokenisation, we can leverage the strengths of transformer architectures and benefit from the advancements made in language modeling. Furthermore, this approach paves the way for unified model architectures for seamless integration of multiple modalities.

Image is high dimensionality modality, for example, a single 8-bits RGB pixel can represent $256 \times 256 \times 256 = 16,777,216$ possible colour combinations. The continuous nature of image data means there are an infinite number of potential variations in color, texture, and spatial arrangements. Representing all this detailed information within a fixed set of image patches would result in an unmanageable complexity.

2.3.5.2 Discrete Image Tokens

To address the shortcomings of continuous image tokens, discretisation is used to quantise these continuous features into a finite set of discrete codes, making the representation more manageable and efficient for both learning and generation tasks while still preserving essential patterns and details. These methods include discrete VAE (dVAE)[94] and its variants such as VQ-VAE[119] and VQ-GAN [26] are employed. The compressed image tokens preserve rich image features by mapping high-dimensional pixel data into a finite set of discrete tokens that capture essential patterns and textures.

During this process, the continuous image data is encoded into a lower-dimensional latent space. The quantisation step selects from a codebook of discrete values, ensuring that the compression maintains the most informative aspects of the image. The image token encoding and decoding are illustrated in Figure 2.4. This allows the model to retain crucial details necessary for high-quality image reconstruction and generation, while reducing the complexity and size of the transformer input.

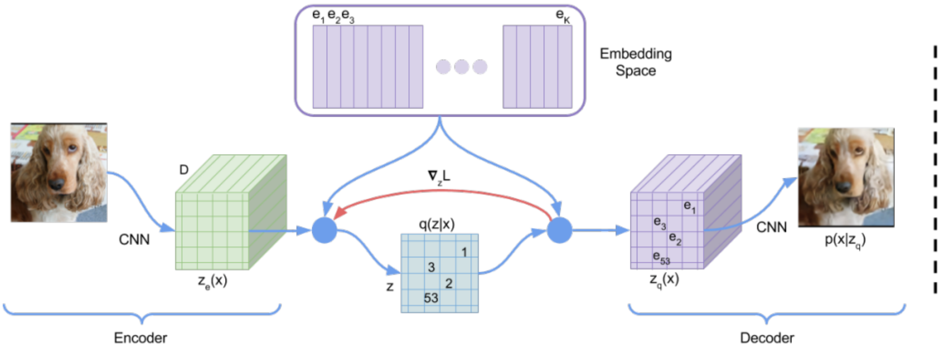


Figure 2.4: Diagram showing latent variables z is quantised into discrete tokens q during training. They can be used to retrieve the latent variable embedding, e to reconstruct the image via the decoder, reprinted from [119].

2.3.6 Image Attention in T2I Transformers

The seminal paper DALL-E [90] re-purposes the transformer architecture for text-to-image generation by adopting both text tokens and image tokens. In training, all the text and image tokens are concatenated to form a single stream of data and modelled autoregressively like LLMs. This means the transformers treats image token in the same way as text tokens, and predict the image tokens in sequence. DALL-E is a decoder-only transformer with self-attention layers in which all image tokens can attend to all text tokens. In this setup, each image token has access to all text tokens, allowing comprehensive cross-modal alignment. However, for efficiency in handling large image data, DALL-E optimizes image-to-image attention by adopting a separated row and column attention mechanism across different layers. This reduces the computational load, which would otherwise be $O(N^2)$ with respect to the number of image tokens N.

For row attention, each token attends to only a limited set of prior tokens in the same row (*e.g.* the preceding 5 tokens), and similarly for column attention. While this localised approach helps manage computation costs, it restricts the model's long-range dependency capability, potentially leading to coherence errors. For instance, if the model generates a human head in the upper section of the image, this attention constraint might prevent it from referencing this information accurately in other parts of the image, possibly resulting in additional, misplaced heads or disjointed features as will be shown in the next chapter.

The attention-mechanism represents a significant departure from traditional conditioning methods, which typically involve concatenating or adding conditioning embedding directly to the model's layer embeddings. Attention dynamically learns how to distribute attention across different parts of the conditioning tokens (both text and image) and selectively integrates that information based on its relevance to specific parts of the image being generated. This results in a more flexible, context-sensitive conditioning process, where the model can refine its output by focusing on relevant features at each generation step, leading to more coherent and detailed outputs.

We adopted transformer architecture in two chapters - Chapter 3 that uses autoregressive training for text-to-image; and Chapter 6 as denoising network in diffusion models for video generation.

2.4 Diffusion Models

Shortly after the introduction of autoregressive transformer for image generation, diffusion models have quickly emerged as a powerful approach for image generation. The development of modern diffusion models traces back to [107] which introduced diffusion probabilistic models (DPMs), where data undergoes a gradual noising process and is learned to be reversed for generation. This idea was refined by [40] with Denoising Diffusion Probabilistic Models (DDPMs) and [22], which improved training and sampling efficiency, leading to state-of-the-art generative performance. In parallel, Song and Ermon [109] introduced score-based generative models, which estimate the gradient (score) of data distribution and denoise samples iteratively. This approach was later unified with diffusion models through stochastic differential equations (SDEs) in [110], providing a flexible framework for training and sampling.

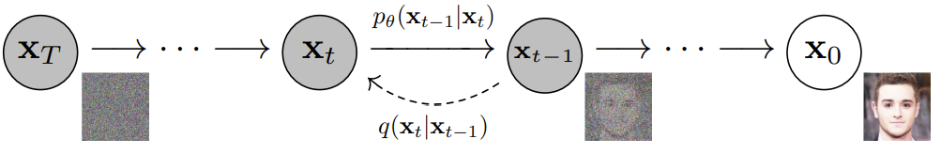

$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$

Figure 2.5: Diffusion process to iteratively produce image from random noise, reprinted from [40].

Modern diffusion models based on DDPM [40], start by adding noise to an image and progressively remove this noise through multiple iterations to recover the target image. In the forward diffusion process in training, noise is gradually added to the data over a series of time steps. This process can be described mathematically as follows. Let x_0 be the original data (e.g., an image), and define a series of noisy observations x_t at time step t:

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon \tag{2.5}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is Gaussian noise added at each step and α is a noise schedule that controls the variance of the noise at each time step. The goal during training is to learn a denoising function θ that can reverse this diffusion process. In practice, θ is modeled with deep neural networks, predominately U-Net although transformers have also gained significant traction recently. In other words, a denoising network is trained to predict the noise ϵ to be used to denoise the noisy image, restoring it to 'clean' image. Hence, the training loss is minimisation of L2 loss between the noise and predicted noise:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0, \epsilon, t} \left[\|\epsilon - \theta(x_t, t)\|^2 \right] \tag{2.6}$$

Once trained, the model can generate new samples by reversing the diffusion process. Starting from pure noise x_t, the model iteratively refine s this noise back into a data sample x_0:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \theta(x_t, t) \right) + \sigma_t z \tag{2.7}$$

where $z \sim \mathcal{N}(0, I)$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ is the cumulative product of α. In the original sampling employing DDPM [40], sampling involves a slow, multi-step Markovian process where noise is gradually added and then reversed over many steps, requiring hundreds of iterations to generate high-quality images. DDIM[108] improves this by using a non-Markovian, deterministic process, reducing the number of steps needed for sampling while maintaining image quality, leading to faster and more efficient inference.

Early diffusion models [40, 79, 99] operate directly in pixel space, which is computationally expensive especially for high-resolution images. To address this, [95] proposed the use of image latents similar to transformer's approach of image tokens, resulting in significant savings in computational resources and enabling higher image resolutions. The latent diffusion model (LDM) has since become the predominant architecture and has been adopted by open source models including the popular T2I model Stable Diffusion [112].

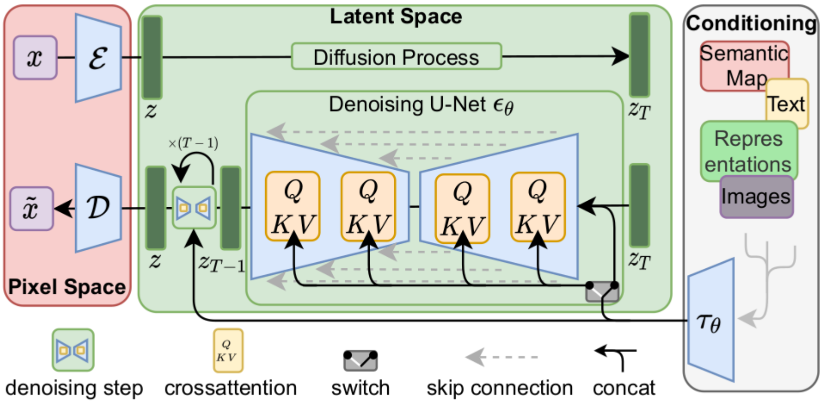


Figure 2.6: τ_θ is modality-specific encoders to extract features embeddings. The conditioning embeddings can then be applied using feature fusion (concatenation) or cross-attention mechanism. Reprinted from [95].

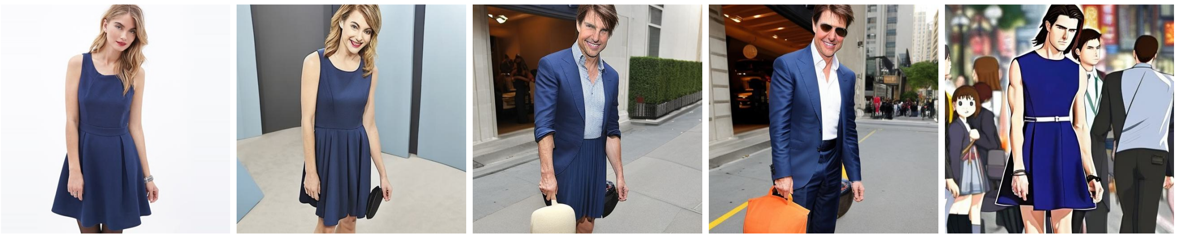
Figure 2.6 shows the architecture of a U-Net based LDM. Unlike the U-Net architecture traditionally used in GANs, which primarily relies on convolutional layers, the U-Net architecture in diffusion models integrates attention layers, hence supports conditioning via cross-attention mechanism. The input conditioning *e.g.* text, images is projected with encoder τ_θ into embedding to go into K and V of cross-attention layers, with Q derived from intermediate features of the U-Net. In this architecture, direct feature fusion is also possible by concatenating with the latent noise z_T. For example, [100] concatenates a low-resolution image to the latent noise, serving as input condition in image super-resolution tasks. This architecture opens doors to the conditioning of different modalities such as parametric pose conditioning pioneered in this work.

2.4.1 Diffuser Adapters

Training large diffusion models is computationally expensive, and adding new conditioning mechanisms after training typically requires costly retraining of the entire model. To address this challenge, adapters can be used as a lightweight alternative. Adapters [44] originated as a lightweight fine-tuning method for LLMs to address the challenge of efficiently adapting pre-trained models to new tasks without retraining the entire network. Instead of modifying the entire model, adapters introduce small task-specific models to extract the feature embedding and applied to the pre-trained models. This allows for the incorporation of new conditioning information without the need for full retraining, significantly reducing both computational costs and time while maintaining the flexibility to adapt the model to new tasks or conditions.

Recently, adapters were introduced to diffusion models. With the pre-trained diffusion models frozen, only the adapters are trained to extract conditioning features *e.g.* sketches, segmentation map, pose skeleton, and then applied to the base diffusion models. The weights of control adapters are usually zero-initialised [143] so the changes is introduced gradually to maintain the pre-trained models capacity. The conditioning features are commonly fused with the pre-trained U-Net internal mutiscale features with spatial addition *e.g.* ControlNet [143], T2I-Adapter [77], making them particularly effective for structural conditioning tasks, such as sketch-to-image and pose-to-image translation. This approach is akin to image-to-image translation in GANs.

On the other hand, image conditioning can be achieved by injecting image features. IP-Adapter [136] performs additional cross-attention with image features from reference image, and add the cross-attention features with the cross-attention with text embeddings in the original T2I diffusion models. The use of cross-attention rather than direct feature fusing, allowing for more spatially flexible conditioning, such as transferring artistic styles across images.



(a) Visual condition **(b)** 100% control **(c)** 50% control **(d)** 20% control **(e)** Our method

Figure 2.7: Combination of IP-Adapter and ControlNet unable to produce desired result of Tom Cruise (text condition) in blue dress (visual condition) in given pose and anime styles (text condition). Our single-brach adapter method can generate harmonious result as shown in (e). Reprinted from [18].

Multiple adapters of different input conditioning can also be applied simultaneously to provide better controllability but they can create conflicts among themselves and with the base image model. This can be illustrated in Figure 2.7. In this example, two adapters are applied to a T2I model - ControlNet for human pose and IP-Adapter for visual conditioning, alongside the text prompt “Tom Cruise in anime style”. The aim of the is to generate an image of Tom Cruise (text) wearing the blue dress (image) in the same pose (pose), and presented in anime image style (text). Without full signal strength coming from the adapter, result in Figure 2.7b shows that the visual condition from IP-Adapter completely overpowered text conditioning applied to the T2I model, resulting in an output dominated by the IP-Adapter’s visual conditioning. To mitigate this, the IP-Adapter’s control strength is reduced 50% by multiplying the control signals by 0.5 before fusing with the T2I model. This restored some text controllability to create an image of Tom Cruise (Figure 2.7c), albeit at the expense of resemblance to the blue dress in reference image. However, futher reduction of the IP-Adapter’s strength (Figure 2.7d) still fails to yield an anime style due to conflicting pose conditioning signals from ControlNet, which was primarily trained on images of real people. This illustrates the adapter branch conflicts leading to mode conflict. Addressing this issue is crucial for achieving harmonious multimodal conditioning. Figure 2.7e shows the desired result as produced using our method, which will be elaborated on in Chapter *Chapter 5: Avoiding Mode*

Conflict with Unified Pose-Visual Diffusion Adapter.

2.4.2 Adapting Image Diffusion Models for Video Generation

Video is composed of a sequence of images, which can be effectively generated using image diffusion models. Rather than training a video diffusion model from scratch on video data, a more efficient approach is to add trainable modules into pretrained image diffusion models to enforce temporal consistency between frames. In the seminal paper, [43] inflates an image diffusion model so that it takes 5D video tensor $x \in \mathbb{R}^{b\times f\times c\times h\times w}$ as input where b and f represent batch axis and frame axis respectively. The tensors are reshaped to $x \in \mathbb{R}^{(b\times f)\times c\times h\times w}$ when going through image layers, treating each frame as an independent image.

Then, temporal transformer is used to process the spatial tensors in temporal dimension to produce motion and frame consistency. At the input of the temporal transformer, the tensor spatial dimension is blended into batch dimension and reshaped to $x \in \mathbb{R}^{(b\times h\times w)\times f\times c}$. By reusing the spatial feature extraction learned from images and focusing training on only the temporal consistency modules, it significantly reduces computational cost and accelerates convergence, making high-quality video generation more accessible.

2.4.3 Motion Control

Later, AnimateDiff [33] demonstrated that the temporal transformer, also known as the motion module or adapter, can be trained separately for different types of motion and then integrated into a base T2I model to generate specific motion patterns. However, this approach lacks fine-grained control, as a separate motion adapter must be trained for each type of motion. To address this, [35, 127] adds camera control by directly fusing the camera conditioning to the frame features before temporal transformer. These methods work well for U-Net based diffusion models but became ineffective when implemented for transformer-based diffusion (DiT). This motivated our research into finding optimal camera conditioning method for video generation in *Chapter 6: Guiding Camera Motion in Video Diffusion Transformer*.

2.4.4 Diffuser Guidance

In addition to feature fusion and cross-attention, *diffuser guidance* emerged as an effective and complementary conditioning method for diffusion models. Inspired by [107, 110], [22] proposed classifier guidance to generate images from class labels. They pre-trained a image classifier on noisy images, and use the gradient the log-probability of a desired class to modify the diffusion trajectory, pushing the generated image towards a specific class. This is captured by the following equation:

$$\tilde{\theta}(x_t, t, y) = \theta(x_t, t) - s\{\nabla_{x_t} \log p(y \mid x_t)\} \tag{2.8}$$

where

- $\tilde{\theta}(x_t, t, y)$ is the adjusted noise prediction for time step t and label y
- $\theta(x_t, t)$ is the original noise prediction from the diffusion model

- s is a scaling factor that controls the strength of the guidance
- $\nabla_{x_t} \log p(y \mid x_t)$ is the classifier gradient

The classifier guidance modifies the noise estimate at each step of the diffusion process, nudging the image generation toward samples that are more likely to belong to the desired class. This results in more controlled and class-specific generation. This method, as adapted by [19], performs text-to-image generation using CLIP-based text guidance [86]. The CLIP model employs separate text and image encoders, which project each modality into a shared embedding space. In the CLIP-guided approach, the cosine similarity between the text embedding and the embedding of the generated images is used to compute the gradient for classifier guidance.

Classifier-free guidance [42], offers several advantages over classifier-guidance and has since become the preferred method for guiding diffusion models. Most notably, it eliminates the need to train a separate classifier, simplifying the overall model architecture and training process. Classifier-free guidance works by training a single model that can perform both unconditional and conditional generation. During training, the model is conditioned on some information (e.g., text, labels) most of the time, and allowed to generate unconditionally for the rest. At inference, the model's output is a combination of these two modes, where the guidance is applied by adjusting the noise prediction. Specifically, the model predicts the noise $\epsilon_\theta(x_t, c)$ conditioned on some information c (e.g., text), and unconditionally $\epsilon_\theta(x_t)$ without the condition. The conditional sample is then guided by guidance scale s, a scalar value that controls the strength of the conditioning, modifying the noise estimate as follows:

$$\hat{\theta}(x_t) = \theta(x_t) + s\{\theta(x_t, c) - \theta(x_t)\}) \tag{2.9}$$

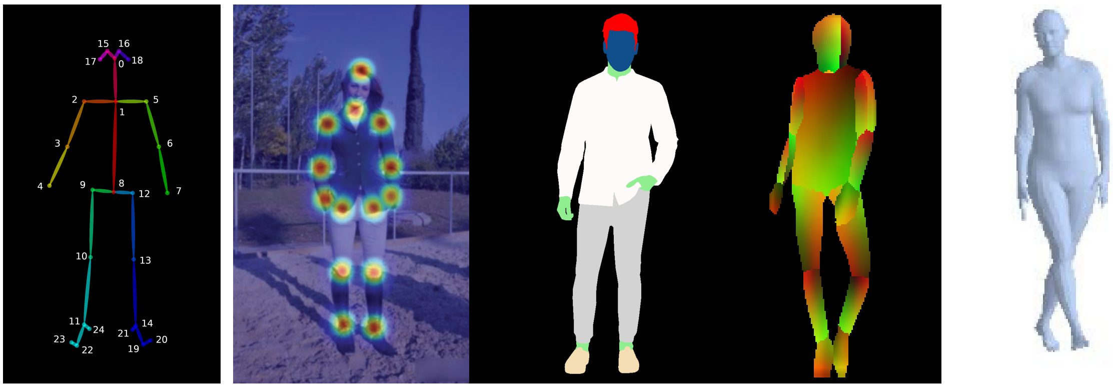
In essence, this approach calculates the direction of the conditional embedding and extrapolates it from the unconditional prediction, guiding the model toward samples that better align with the conditioning input. Classifier-free guidance has been widely adopted for text guidance in diffusion models to generate images [27, 28, 82–84, 89, 95, 112, 143] and videos [39, 43, 73, 75, 134, 135, 148]. In *Chapter 6: Guiding Camera Motion in Video Diffusion Transformer*, we enhance classifier-free guidance to address the challenges of ineffective camera motion control in transformer-based video diffusion models.

2.5 Pose Representations

Human and camera pose representations play a vital role in the creative processes of filmmaking, animation, and visual storytelling. Human pose, for instance, conveys the body's positioning and movement, while camera pose defines the viewpoint and framing. Capturing and controlling these poses is essential for accurately expressing creative intent. Text prompts alone are insufficient to describe them with the precision needed for such tasks. In this thesis, we will explore parameterized methods to control pose. Now, we provide an overview of existing and common pose representation.

2.5.1 Human Pose

In the context of computer vision and machine learning, keypoints are used to represent the structure of the human body. Body keypoint refer to specific anatomical landmarks on the human body, such as joints (e.g., elbows, knees) and other significant points (e.g., shoulders, hips, ankles). A typical human body model might represent anywhere from 14 to 25 keypoints depending on the complexity, capturing both upper and lower body joints. These keypoints can be detected from 2D images using models such as OpenPose [12] to produce the Cartesian coordinates in 2D (x,y) or 3D (x,y,z) where the additional z-coordinate represents depth information.



(a) Skeleton image **(b)** Heatmap **(c)** Segmentation **(d)** DensePose **(e)** SMPL

Figure 2.8: Illustration of various human pose representation methods.

GANs rely on convolution layers for feature extraction. However, convolutional layers are designed to capture local spatial patterns, which makes them less suited for detecting globally distributed or sparse features, such as the scattered joints in body keypoint detection. Therefore, early methods convert the keypoints into 2D spatial representation of heatmaps $\mathbb{R}^{H\times W\times K}$ where H, W denote the image dimension and K represents the number of keypoints, *i.e.* one feature map per keypoint. Subsequently, skeleton images—depicting lines that connect keypoints—gained popularity, as they can be represented as RGB images with only three dimensions, fitting seamlessly into the image-to-image translation frameworks of GANs. However, keypoints alone do not capture body shape information, prompting the development of pixel-level dense body representations, including 2D body segmentation maps and 3D surfaces, such as DensePose [34]. In a segmentation map, each pixel is assigned a class label—such as hair, face, shoes, and shirts—with distinct colours used for visualisation. In contrast, DensePose provides a dense mapping of image pixels to a 3D surface model of the human body. Typically, both representations are presented as 2D images [27, 51, 81] to serve as conditioning inputs for generative models.

One limitation of these human poses extracted from human images are that they are not easily modifiable. Changing one keypoint often requires adjusting others to maintain a realistic body structure and ensure that the proportions and angles between limbs remain anatomically plausible. This interconnectedness makes it difficult to manually modify keypoints without inadvertently distorting the pose. It is even more challenging for editing pose in dense representation. Therefore, parameterised 3D body models such as Skinned Multi-Person Linear model (*SMPL*) [69] has been developed to address this problem. SMPL model is a parametric human body that captures the complexities of human body shape and pose using separate body pose and body shape parameters. The body pose in SMPL is represented by 3 rotation angles of 24 body joints, leading to a total of 72 parameters. In addition, SMPL also incorporates body shape parameters to define the overall shape and size of the human body. These parameters are derived from a principal component analysis (PCA) of a training dataset of 3D body scans, enabling the model to capture a variety of body shapes and proportions. Typically, the body shape is represented by a vector of 10 parameters, allowing the model to adjust for variations in height, weight, and body composition. Figure 2.8e shows an image rendered from SMPL parameters. By leveraging SMPL as a human pose representation, we were the first to investigate a 3D parameterised model for controlling human and camera poses in image generation [17].

2.5.2 Camera Pose

In computer vision and graphics, understanding camera parameters is crucial for accurately modeling the interaction between a camera and the scene it captures. These parameters can be categorised into extrinsic and intrinsic parameters.

2.5.2.1 Camera Parameters

Extrinsic parameters define the position and orientation of the camera in the 3D world. These parameters relate the world coordinates to the camera coordinates and are represented using a rotation matrix $R \in \mathbb{R}^{3\times3}$ and a translation vector $T \in \mathbb{R}^3$. A rotation matrix can be constructed using roll, pitch, and yaw angles, which represent rotations about the X, Y, and Z axes, respectively. The roll (ϕ) corresponds to rotation about the X-axis, the pitch (θ) corresponds to rotation about the Y-axis, and the yaw (ψ) corresponds to rotation about the Z-axis. The individual rotation matrices for each axis are defined as follows:

- Roll (rotation about the X-axis):

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{2.10}$$

- Pitch (rotation about the Y-axis):

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2.11}$$

- Yaw (rotation about the Z-axis):

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

To obtain the combined rotation matrix R, we multiply these individual matrices in the order of yaw, pitch, and roll:

$$R = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi) \tag{2.13}$$

Calculating this product gives the overall rotation matrix:

$$R = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) - \sin(\theta)\sin(\phi)\cos(\psi) & \sin(\theta)\sin(\psi) + \cos(\theta)\sin(\phi)\cos(\psi) \\ \sin(\theta) & \cos(\theta)\cos(\phi) & -\cos(\theta)\sin(\phi) \\ -\sin(\phi) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{2.14}$$

This matrix R represents the rotation of a point in 3D space based on the roll, pitch, and yaw angles. Each entry in the matrix describes how the coordinates are transformed when these rotations are applied sequentially.

Intrinsic parameters define the internal characteristics of the camera, which determine how the camera projects a 3D point in the world onto a 2D image plane. They include focal length, optical center, and any distortion coefficients. The intrinsic matrix K can be represented as follows:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.15}$$

where: - f_x and f_y are the focal lengths in pixels along the x and y axes, respectively, - (c_x, c_y) is the optical center (the principal point) in the image plane. The projection of a 3D point $P = (X, Y, Z)$ onto the image plane can be calculated using the following equation:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.16}$$

where (u, v) are the pixel coordinates of the projected point in the image.

2.5.3 Plücker Coordinates

Plücker coordinates provide a way to represent lines in three-dimensional space using a six-dimensional vector. This representation encapsulates both the direction and position of a line, making it particularly useful in various applications in computer vision and graphics, such as camera calibration, 3D reconstruction, and geometric reasoning. Compared to using extrinsic parameters, Plücker coordinates are invariant under certain transformations, such as perspective projections. This property allows for more robust manipulation and reasoning about lines in 3D space without needing to account for camera motion or orientation changes explicitly.

The Plücker coordinates represent a line in three-dimensional space using a pair of vectors: a direction vector and a moment vector. Given two points on the line, $\mathbf{A} = (x_1, y_1, z_1)$ and $\mathbf{B} = (x_2, y_2, z_2)$, the derivation proceeds as follows:

1. Direction Vector: The direction vector **d** is computed as:

$$\mathbf{d} = \mathbf{B} - \mathbf{A} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix} \tag{2.17}$$

2. Moment Vector: The moment vector **M** can be expressed using the cross product of the position vectors from the origin to points **A** and **B**:

$$\mathbf{M} = \mathbf{A} \times \mathbf{B} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \times \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} y_1 z_2 - z_1 y_2 \\ z_1 x_2 - x_1 z_2 \\ x_1 y_2 - y_1 x_2 \end{bmatrix} \tag{2.18}$$

3. Plücker Coordinates: The Plücker coordinates **L** are then defined as:

$$\mathbf{L} = \begin{bmatrix} M_x \\ M_y \\ M_z \\ d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} y_1 z_2 - z_1 y_2 \\ z_1 x_2 - x_1 z_2 \\ x_1 y_2 - y_1 x_2 \\ x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix} \tag{2.19}$$

Thus, the Plücker coordinates provide a compact representation of a line in 3D space, capturing both its direction and the moment associated with it. Recently, it has been shown [35] to perform better than extrinsic parameters in controlling camera pose for video generation.

2.6 Datasets

Two datasets are used in this thesis. DeepFashion dataset [156] is a human image dataset and is used in Chapter 3 - 5. On the other hand RealEstate10K [149] is used in Chapter 6 for video camera pose.

2.6.1 DeepFashion

The DeepFashion dataset (available on `https://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html`) is a widely-used collection in fashion and human image generation research, particularly valued for its high-quality human images. Each image typically features a single model posing against a plain studio background as shown in Figure 2.9. The dataset comprises around 800,000 images, organised into several categories with varying image resolutions (from 256×176 to 1101×750 pixels) and annotations specific to different tasks such as text captions, fashion labels, parsing maps, and keypoints.

Figure 2.9: Image samples from DeepFashion dataset.

In our experiments, we primarily utilised the images from the DeepFashion dataset and applied a series of tailored pre-processing steps to obtain the necessary information using tools listed below:

- 2D body pose - OpenPose [12]
- 3D SMPL body pose and silhouette mask - Phosa [141]
- Body parts segmentation - [64]

2.6.2 RealEstate10K

The RealEstate10K dataset (available on `https://google.github.io/realestate10k`) is a large-scale collection of real estate video footage, primarily used in research areas like view synthesis, video generation, and 3D scene understanding. It contains approximately 80,000 YouTube videos of real estate tours, capturing a wide variety of indoor scenes from different homes, including rooms, corridors, and architectural details. Each video contains a scene of varying length and each frame is provided with camera intrinsics and extrinsic parameters. This makes RealEstate10k particularly beneficial for camera control task.

2.7 Summary

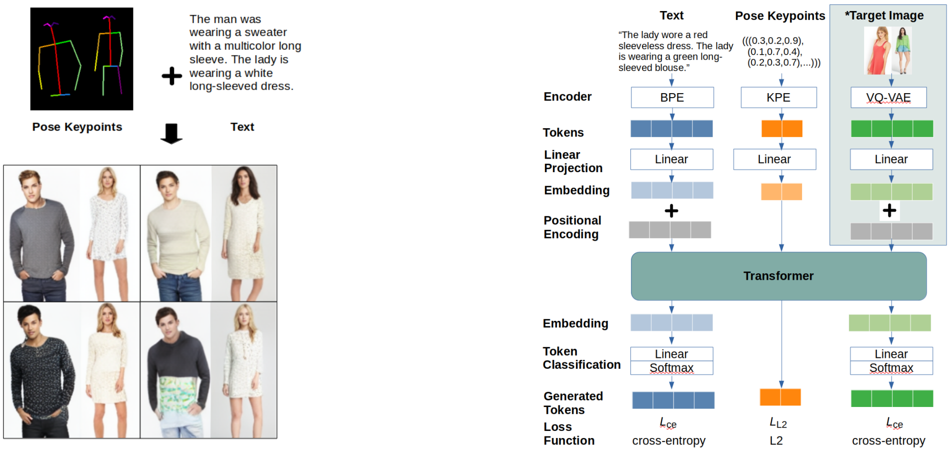
In this chapter, we explore the evaluation of model architectures for image generation and the development of conditioning methods that enhance their functionality. We began with early GANs, which employed simple layers to project conditioning embeddings for direct feature fusion through concatenation or addition. With the advent of transformers, attention mechanisms, particularly cross-attention, became widely adopted, enabling models to better align and integrate information from various modalities. More recently, lightweight adapters have been introduced to fine-tune or add additional controls to pre-trained large diffusion models, leading to a proliferation of creative applications leveraging diffusion adapters. Additionally, diffuser guidance, particularly classifier-free guidance, has been applied to diffusion models to modulate the strength of conditioning. Finally, both human and camera pose representations were introduced, which will be used in our work as conditioning to complement the existing text and image modalities in the models.

Chapter 3

Parametric Human Pose Token for Autoregressive Transformer

Autoregressive transformer was the de-facto architecture for text-to-image before diffusion models rising to prominence. Hence, we leveraged this architecture to develop the first image generation model using multimodal inputs of text and pose. Code is available on `https://github.com/soon-yau/kpe/`

3.1 Introduction



(a) (b)

Figure 3.1: (a) Our pose constrained text-to-image model supports partial and full pose view, multiple people, different genders, at different scales. (b) The Architectural diagram of our pose-guided text-to-image generation model. The text, pose keypoints and image are encoded into tokens and go into an transformer. *The target image encoding section is required only for training and is not needed in inference.

Autoregressive transformer *e.g.* DALL-E[90] marked a significant improvement from GANs for text-to-image generation. Despite advancement in using text prompt to influence the image outcome, it is still difficult to describe in words entirely the body shape, size, pose, clothing details, position in the images and camera view. Thus generating and controlling human images remains a challenging task. Hence, we propose to add human pose as an additional input to text, to improve the accuracy and fidelity of people being generated. We can see this as enforcing disentanglement of content and style of image [30] where the content is the pose, and the text depicts the style. Concurrent to our work, Text2Human [51] uses text and pose as input, but their method involves complex and specialised neural networks, compared to generic transformer architecture that we use.

The standard method of representing pose for transformers, e.g. [27] is to convert a skeleton image into discrete image tokens with a dVAE. This is the method adopted by VQGAN [26] which frames pose-to-image as an image-to-image problem. Figure 3.2 illustrate how a 256×256 skeleton image is encoded into discrete tokens for transformer. First, the skeleton image is divided into a grid of say 16×16 image patches each containing 16×16 pixels. Then the 256 image patches are encoded using discrete VAE encoder into image tokens, and *flatten* as transformer expect 1-D vector of tokens as input. One major drawback of this approach is its computational inefficiency, as the complexity of the attention mechanism grows quadratically with the number of image tokens. Skeleton image tokens exacerbate this issue, as they often include significant redundancy, with many tokens representing background elements rather than conveying essential pose information. Although separating row and column attention, as demonstrated in DALL-E [90] and axial attention mechanisms [41], can mitigate some of the computational burden, much of this overhead could be avoided by compressing pose information into a single token.

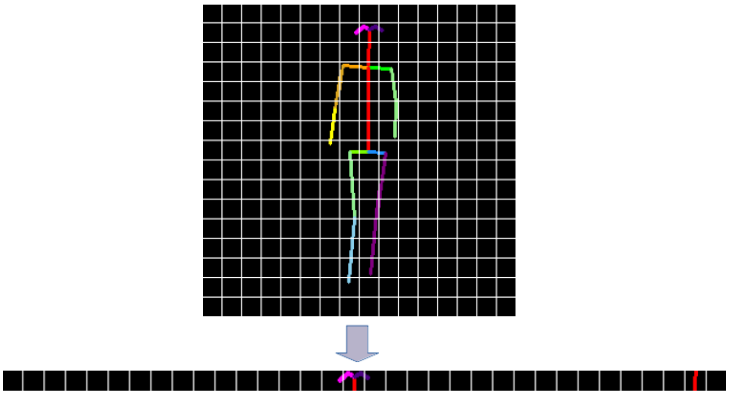


Figure 3.2: Skeleton image token encoding - image is divided into grid, tokenised and flatten.

Inspired by these observation, we have devised **Keypoint Pose Encoding** (KPE) - a novel, efficient and accurate pose representation suitable for a transformer. Instead of using the high dimensional skeleton image mainly containing redundant information, we focus on only the body joint keypoints for pose representation. The low dimensional representation is invariant to changing the target image resolution or domain, e.g. from the natural landscape to synthetic objects. We show it to be 10× more memory efficient and increase computational inference speed by over 73% in the experiment section.

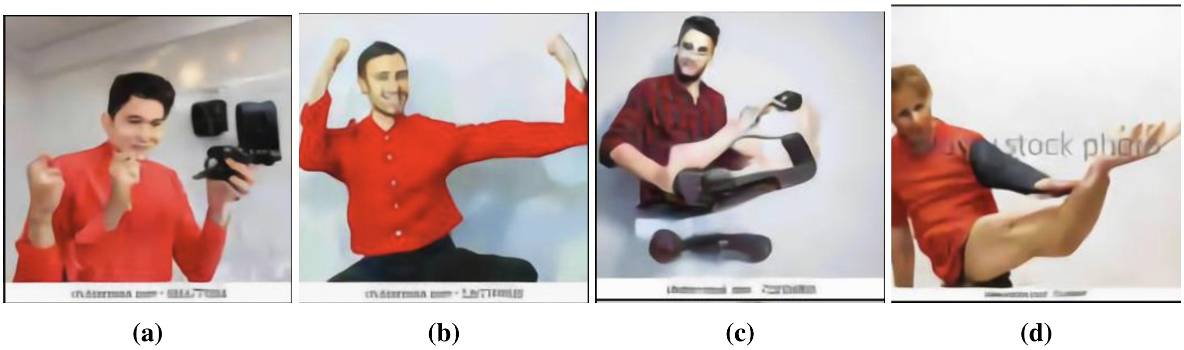


Figure 3.3: Common person-specific errors in images generated by then-state-of-the-art model Cogview (Figure 6 in [23])

To measure success and motivated by the inadequacy of existing metrics to measure image errors specific to people, we devised the **People Count Error** (PCE) metric to measure the false positive rate when generating images of multiple people. Therefore, we can both empirically and qualitatively show that the disentanglement improves the fidelity of people generated with a significantly reduced number of false positive or erroneous body parts.

In summary, our key contributions are:

1. **Novel Keypoint Pose Encoding (KPE)** To enable the *tokenisation* of a human pose representation that is computationally efficient and invariant to changes in target images such as the resolution. This spearheaded the use of pose tokens, paving the way for parametric control in generative models.

2. **Introduction of a person centric metric** A new metric to measure the false error rate of generated humans in multiperson images in rendered images. This metric helps to illustrate that the disentanglement of pose and text leads to better higher quality results with reduced false positive humans.

3. Successfully demonstrated **Multimodal conditioning** of text and pose inputs with transformer architecture.

3.2 Related Work

3.2.1 Text-to-Image Generation

Most GAN-based text-to-image models such as StackGAN[139], AttenGAN[131], DM-GAN[153], DF-GAN[113] and XMC-GAN[138] are forms of conditional GAN [76] where the text sequence is projected to an embedding vector as a conditioning feature, i.e. the text is modelled as continuous variables. However, overall, GANs have been less successful in T2I generation, leading researchers to shift their focus toward autoregressive transformers pioneered by DALL-E [90], which forms the basis of many modern transformer-based text-to-image models such as CogView[23], NÜWA[129] and Make-a-scene[27]. We employ a similar method to tokenise the text in our models but condition our transformer on a further additional modality, *i.e.* pose, to enrich the image quality and precision. On the other hand, there emerged diffusion models *e.g.* GLIDE[79], DALL-E 2[89] and Imagen[99]. We experimented with diffusion models with pose tokens created using our KPE method and it has shown to work. However, the computational benefit is more profound for transformer hence our focus.

3.2.2 Pose Guided Image Generation

There are a few existing methods that represent pose in the context of image generation; including, body keypoint heatmaps [72][105][133], segmentation maps [81][27] and a skeleton image [48][154]. The pose-to-image is framed as image-to-image, where an input image is a form of 2D spatial tensor representing pose, and the output image is the person image. However, these representations include the redundant background in addition to the foreground segmentation or skeleton data. The reason for using the 2D spatial tensor for pose representation is that the spatial information is required for convolutional layers in GANs [116], but this is no longer a pre-requisite for transformer. Despite this, recent generative transformer models [26][129][27] continue using an image for pose representation by encoding image into discrete image tokens. The image tokenisation process can be very slow, [20] attempts to reduce the training time by pre-encoding the images into tokens, but this prohibits the use of augmentation onto the images and poses during training. Thus, several papers [50, 63] realised the shortcoming of using skeleton images and started using keypoint for pose estimation regression. However, we are the first to use pose keypoint to guide the image generation.

3.3 Method

Figure 3.1(b) shows the overall architecture of our pose constrained text-to-image model. The first stage is to convert the text, pose keypoints and image into tokens with their respective encoders. Then the tokens are projected into an embedding space before adding positional encoding. We use learnable positional encoding for text tokens and axial positional encoding [41] for image tokens due to its 2D structure. We do not use positional encoding for keypoint tokens as they are equally important for all positions of the image tokens. Our model is adapted from [122] which is a decoder-only transformer with 12 transformer blocks. We will now describe the details of the token encoders.

3.3.1 Text Encoder

Like DALL-E [90], we use the BPE (Byte Pair Encoding) tokeniser [103] for text tokenisation, the encoder breaks the word into subwords, and they are assigned discrete identifier numbers, which become the text tokens. The text tokens are then projected into embedding $\mathbb{R}^{t \times d}$, where t is the fixed text token input length and d is the transformer dimension. The BPE tokeniser is pretrained with the vocabulary of the dataset.

3.3.2 Image Encoder

In our model, we use VQ-VAE [119] from VQGAN [26] for image encoding and decoding. Its GAN training pipeline has produced better image quality than the dVAE used by [90]. The encoder E first converts the continuous image $x \in \mathbb{R}^{H \times W \times C}$ with height H, width W and colour channel C into code $\hat{z} = E(x) \in \mathbb{R}^{h \times w \times n_z}$ with h and w the spatial dimension and n_z the dimension of the code. Then, each of the codes is quantised $\mathbf{q}(.)$ to its closest discrete codebook entry z_k using the equation:

$$z_q = \mathbf{q}(\hat{z}) := \left(\arg\min_{z_k \in Z} \|\hat{z}_{ij} - z_k\| \right) \in \mathbb{R}^{h \times w \times n_z} \tag{3.1}$$

where $Z = \{z_k\}_{k=1}^{K} \in \mathbb{R}^{n_z}$ is the discrete codebook. The decoder G reconstructs the discrete code into image $\hat{x}$:

$$\hat{x} = G(z_q) = G(\mathbf{q}(E(x))) \tag{3.2}$$

The model and codebook can be trained end-to-end using the loss function:

$$L_{VQ}(E, G, Z) = \|x - \hat{x}\|^2 + \|sg[E(x) - z_q]\|_2^2 \tag{3.3}$$

$$+ \|sg[z_q] - E(x)\|_2^2 \tag{3.4}$$

where $sg[.]$ denotes stop-gradient operation.

Overall, the image is tokenised into $h \times w$ grid of discrete image tokens. Although h and w are hyperparameters, they have a linear correlation with image resolution H and W to maintain the same quality of image texture details. Therefore, increasing image resolution will lead to more extended image token lengths, resulting in a quadratic increase in computational complexity. The image encoder is pretrained with the target images. The discrete image tokens are then projected into transformer embedding $\mathbb{R}^{h \times w \times d}$.

3.3.3 Keypoint Pose Encoder (KPE)

KPE is our method for pose representation. It converts pose positions for multiple people into keypoint tokens and then encodes them into a keypoint embedding.

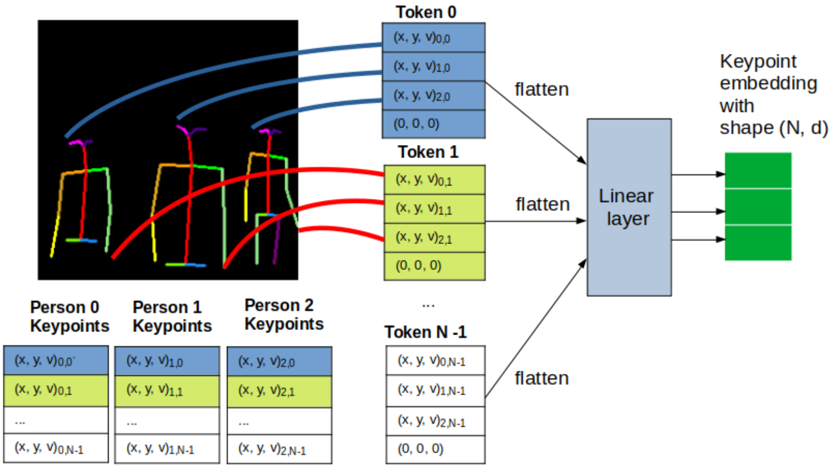


Figure 3.4: The Block diagram showing KPE encoding multiperson pose to keypoint tokens. The tokens are flattened and projected into keypoint embedding within the transformer. The skeleton image is for illustration purposes as we use the keypoints directly from the pose estimation model's outputs.

A single 2D keypoint is defined as a tuple of (x, y, v) where x and y are the normalised Cartesian coordinates in [0, 1] and v is the visibility score in $[0, 1]$. We denote multiperson 2D keypoint format as $(x, y, v)_{i,j}$ where i is the person index, j is the keypoint index from 0 to $N - 1$ where N is the total number of keypoint defined. Different pose estimation models use different keypoint schemes, but the common keypoints are the nose, neck, shoulder, elbow, wrist, hip, knee, ankle, eye, ear, big toe, and heel.

Figure 3.4 shows the process of KPE. The process is similar to the method in Vision Transformer (ViT)[24] of grouping image pixels into image patches, except that we group keypoints into keypoint tokens. Each keypoint token corresponds to a skeleton joint; in this illustration, keypoint token 0 is designated for the right eye and keypoint token 1 is for the left wrist. The right eye keypoint for all the people in the image are stacked together and moved into token 0. In this example, we define the system to support up to four people, and if there are fewer than the maximum number of people, they will be padded with zeros in the keypoints. The same goes for keypoints that are not visible. The resulting keypoint tokens will have a length of N. In other words, the number of fixed with the keypoint definition does not change with the number of people in the image or changes in image resolution. It is worth noting that the keypoint token is not discrete; it is continuous in the range [0, 1].

The next step is to ensure that the pose embedding has the same dimension as the transformer embedding. We propose two methods; the first one is to pad the keypoint tokens with zeroes to match the transformer dimension. This method is the fastest as it does not require any arithmetic computation. It can accommodate many people up to constraint within $3M <= d$ where M is the maximum number of people, and d is the transformer dimension. We tested this method to be working faster. However, we use a linear layer to project the keypoint tokens into keypoint embedding for the generality of the unbounded number of people. Overall, the KPE converts multi-person keypoints $\mathbb{R}^{M \times N \times 3}$ into embedding $\mathbb{R}^{N \times d}$.

3.3.4 Training

To train, the text tokens T, keypoint tokens K and image tokens I are concatenated to be fed into the transformer. The transformer output has the same length as the input, aiming to generate the same tokens as the input tokens. As the text tokens and image tokens are discrete, prediction of them becomes a multiclass classification problem, and we use the cross entropy loss $\mathcal{L}_{ce}$ as is common with transformer training. However, the keypoint tokens are continuous values, and we use an $L2$ loss $\mathcal{L}_{L2}$ on the keypoint embedding. Therefore, the overall loss function is:

$$\mathcal{L} = \mathcal{L}_{ce}(T) + \lambda_I \mathcal{L}_{ce}(I) + \lambda_K \mathcal{L}_{L2}(K) \tag{3.5}$$

λ_I and λ_K are constants. Higher value encourages more accurate people image and poses, respectively, and their values are discussed in Section 3.4.3.

3.3.5 Inference - Autoregressive Sampling

The inference is performed by repeatedly sampling the next image token, conditioned on text tokens, pose tokens and previously sampled image tokens. The inference starts by fixing the text and pose tokens and run a forward pass to generate score for image token. If image token with the highest probability is selected, the model will generate the same image every time and become deterministic. Therefore, to introduce variation in generating images by randomly sampling from several image tokens the highest softmax probability. The size of the sampling pool is a hyperparameter; setting it too low can give more variation but may also introduce more errors in the image. The sampled image token is concatenated with text and pose tokens to generate the next image token. The process repeats until all image tokens have been generated, then they will be decoded to generate an image.

3.4 Experiments

We tested our idea by training a model on mannequin dataset [90]. The original dataset contains image of a single mannequin with accompanied text description. OpenPose [12] pose estimation was used to extract the pose keypoints. We randomly combine two samples to form a multi-mannequin dataset.

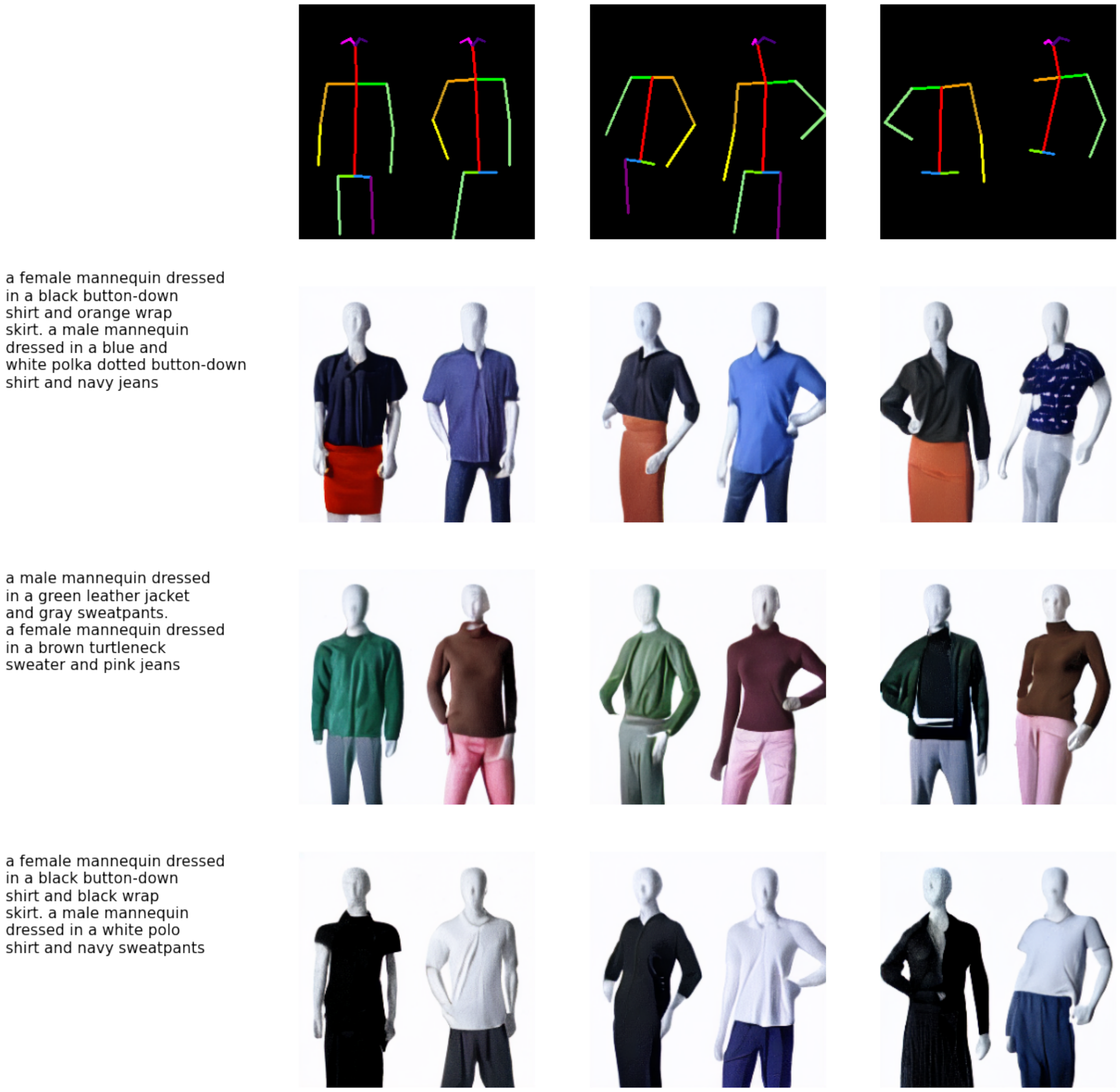


Figure 3.5: Initial results with our model trained on mannequin dataset.

Results in Figure 3.5 shows KPE can deliver accurate pose accuracy with the generated mannequins matching the gender and clothing apperance in the text prompt.

Then we moved on to train a photorealistic DeepFashion [156] dataset for quantitative evaluation with two baseline models that we implemented. We first re-implemented a text-only model DALL-E [90] as an ablation study to understand the effect of adding pose guidance on generated image quality. Then, we added a image pose conditioning from VQGAN [26] to DALL-E[90], to compare the performance of KPE against image conditioning method of using skeleton images for pose representation. We highlight the advantages of KPE in Section 3.5.2.

3.4.1 DeepFashion Dataset

We use DeepFashion's fashion synthesis benchmark dataset for the experiments. The original dataset contains 78.5K images of a single person and a brief description of the gender, clothing colour, and type.

We derived a multiperson dataset by randomly sampling the single-person images, randomly resized them by 10%, cropped them, and concatenated them into a single 256 × 256 image. The background of the individual images is not removed before the concatenation. The new images have between 1 to 3 non-overlapping people in various locations, sizes and poses.

3.4.2 Evaluation Metrics

We evaluate the performance of our approach on two aspects: the similarity or faithfulness of the generated images to text description and pose; and the image quality, which includes the realism of people.

3.4.2.1 Similarity

We use Object Keypoint Similarity (**OKS**) from the MSCOCO keypoint challenge [78] for keypoint accuracy. We also use Structural Similarity (SSIM) to compare the similarity of generated images to the reference images. Like [72] we mask out the background, but we also we crop away the excessive background to avoid SSIM being dominated by the background to give the metric **Mask-SSIM**. We also use **CLIPSIM** [128] to measure the similarity between text inputs and generated images.

3.4.2.2 People Unrealism

To evaluate the *realism* of the images, we use the perceptually trained Inception Score (**IS**) [101] and **FID** [37], in line with literature image generative models [72][154][90][129][23]. However, as suggested by [6], IS sub-optimal, and we found that it is not good at measuring people-centric errors such as missing or extra limbs. To overcome this limitation, we propose a new evaluation metric People Count Error (**PCE**) to measure the unrealism of the people. Given an image of people x, $gt()$ indicates the ground truth function that returns the labelled number of people in the image, and $h()$ is the function that returns the number of people detected by the pose estimation algorithm e.g. OpenPose [12]. Therefore, PCE is defined as:

$$PCE(x) = \begin{cases} 1, & \text{if } h(x) \neq gt(x) \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

PCE makes use of the rich body anatomy knowledge embodied in OpenPose. If a person in a erroneous generated image has three arms, OpenPose know human only has two arms, and therefore it will assume the third arm belong to another person, hence adding the person count. This discrepancy in people count is flagged by PCE as 1 (contain error). Visual examples of PCE are shown in the results in Fig 3.12 and further discussed in Section 3.5.3. Moreover, unlike IS/FID which requires a large amount of data, PCE applies to a single image, making it usable to find an error in an individual image.

3.4.3 Implementation details

We adopt a two-stage training process like [90]. The first stage trains VQ-VAE using the VQGAN pipeline on the target images to encode 256 × 256 images into 16 × 16 = 256 image tokens, where each token can assume 8192 possibilities. To compare against [90], we use an open-source implementation [122] with a transformer dimension d of 512, with 8 heads and a depth of 12 encoder blocks. The text token length is 256, and the input text tokens will be truncated if they exceed this length. We train using the loss function in Equation 3.5 using OpenPose's BODY_25 [80] pose format. Therefore, the keypoint token length is 25, corresponding to the 25 keypoints.

The DALL-E+VQGAN model is also a text-and-pose guided model. The difference with our KPE model is that it uses VQGAN's pose representation method of using VQ-VAE to encode skeleton images into pose image tokens. Like VQGAN, we reuse the VQ-VAE pretrained on target images. The DALL-E+VQGAN's loss function is:

$$\mathcal{L} = \mathcal{L}_{ce}(T) + \lambda_I \mathcal{L}_{ce}(I) + \lambda_K \mathcal{L}_{ce}(P) \tag{3.7}$$

where P are the pose image tokens.

We use the same VQ-VAE, training configuration and hyperparameters for all three models. Therefore, we chose a smaller VQ-VAE, which may not produce the most visually pleasing image quality, but this presents a fair comparison and ablation study. For loss constants, We use λ_I=7 from [122]. We tried 1 and 10 for λ_K but did not notice much difference in the results. Eventually, we select λ_K=10 to have the same order of magnitude as λ_I. For the optimiser, we use Adam [56], with initial learning rate of 0.0001, β_1=0.9, β_2=0.999. The learning rate is reduced by half if the loss has plateaued for 12 epochs until it reaches 1e-6. We use a batch size of 10 and train for 100 epochs on an RTX5000 GPU with 16GB GPU memory.

Due to the non-deterministic nature of image generation, we compute 5 images per sample in the test dataset and obtain the mean value of the metrics for the qualitative result. We sample image tokens from the top 0.1% highest probability or 8 out of 8192 tokens. This sampling improves the image quality and consistency of metrics values.

3.5 Results

3.5.1 Qualitative Results

We present images generated with KPE pose guided text-to-image model in Figure 3.6 - 3.10. All images were generated by our model, using only keypoints (skeleton image is only for illustration purpose) and the text as shown in the figure's caption. This demonstrates our method can generate images that are from pose and text conditions.

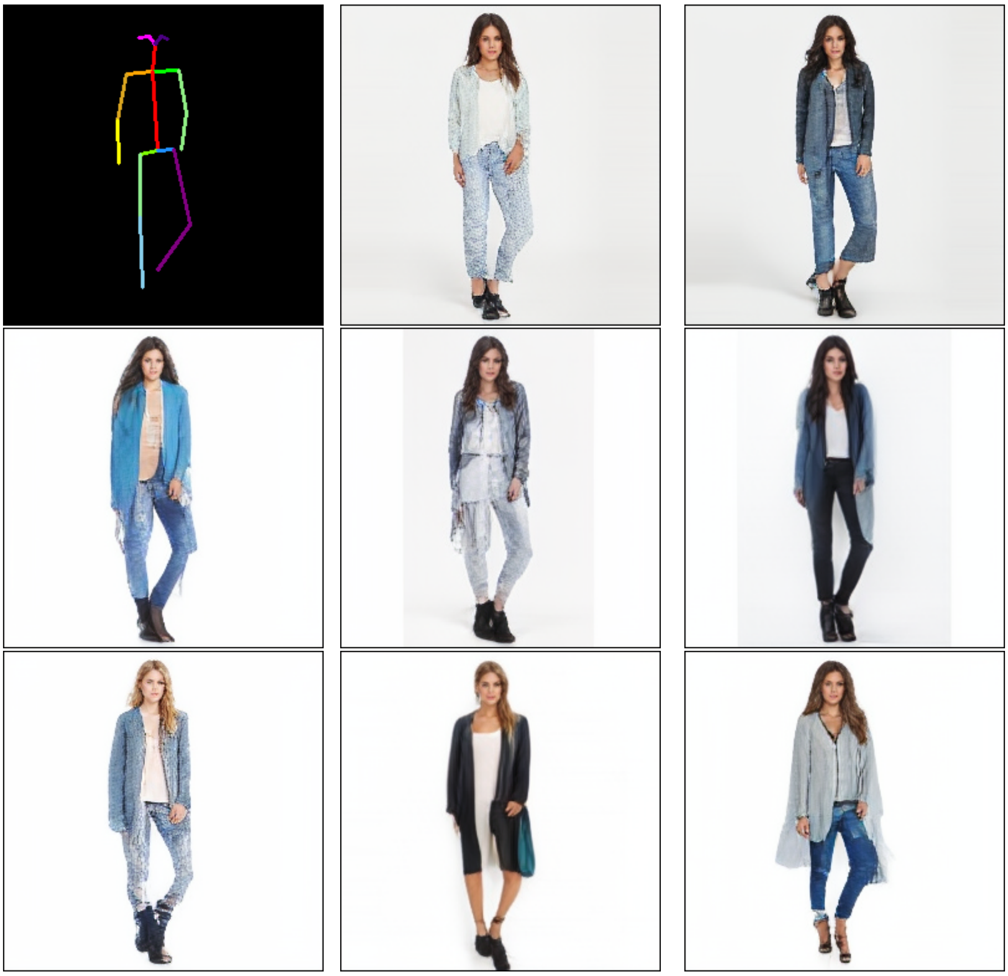


Figure 3.6: ‘the lady wore a blue long-sleeved cardigan.’

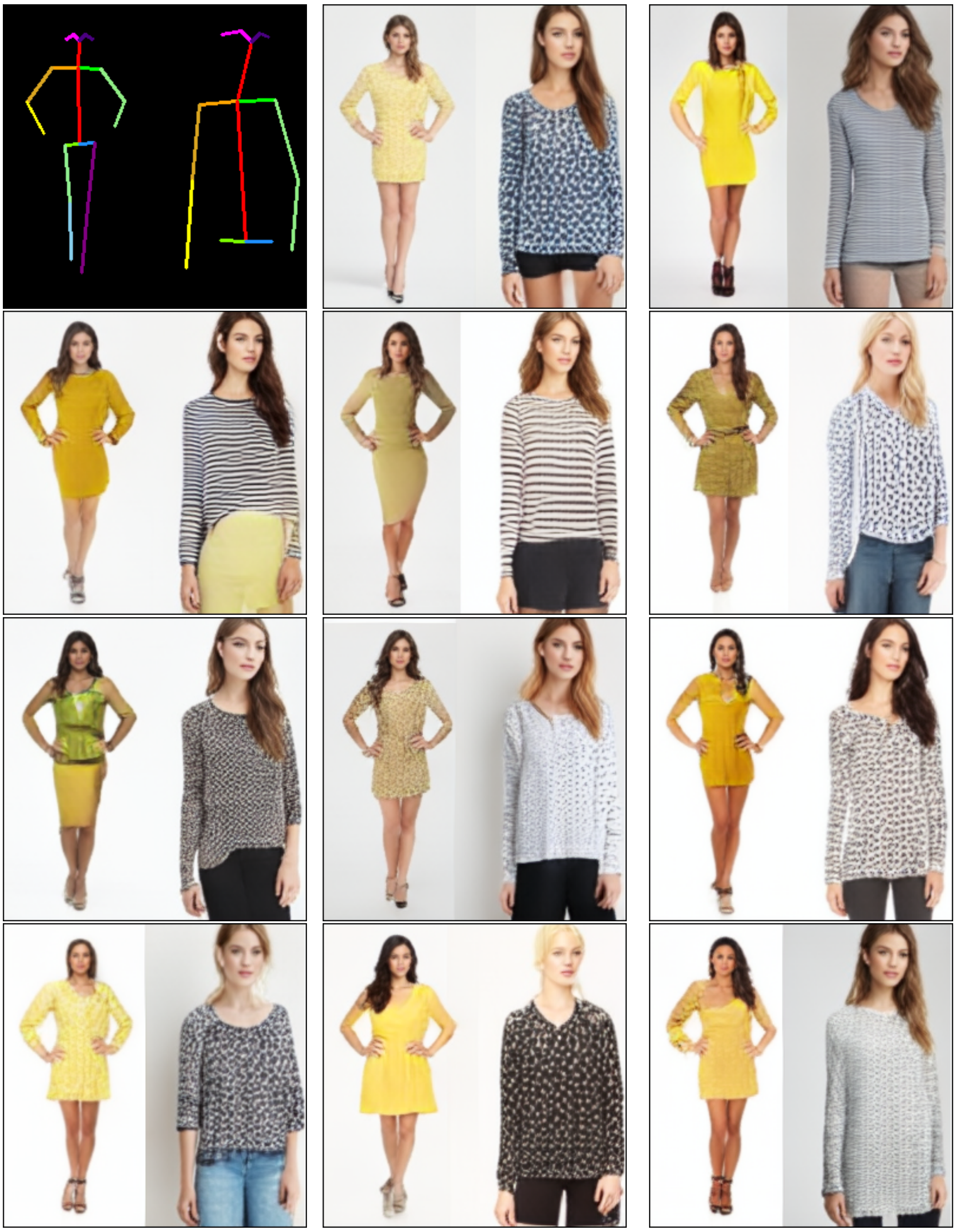


Figure 3.7: 'the lady is wearing a yellow long-sleeved dress.the lady is wearing a multi-colour long-sleeved tee.'

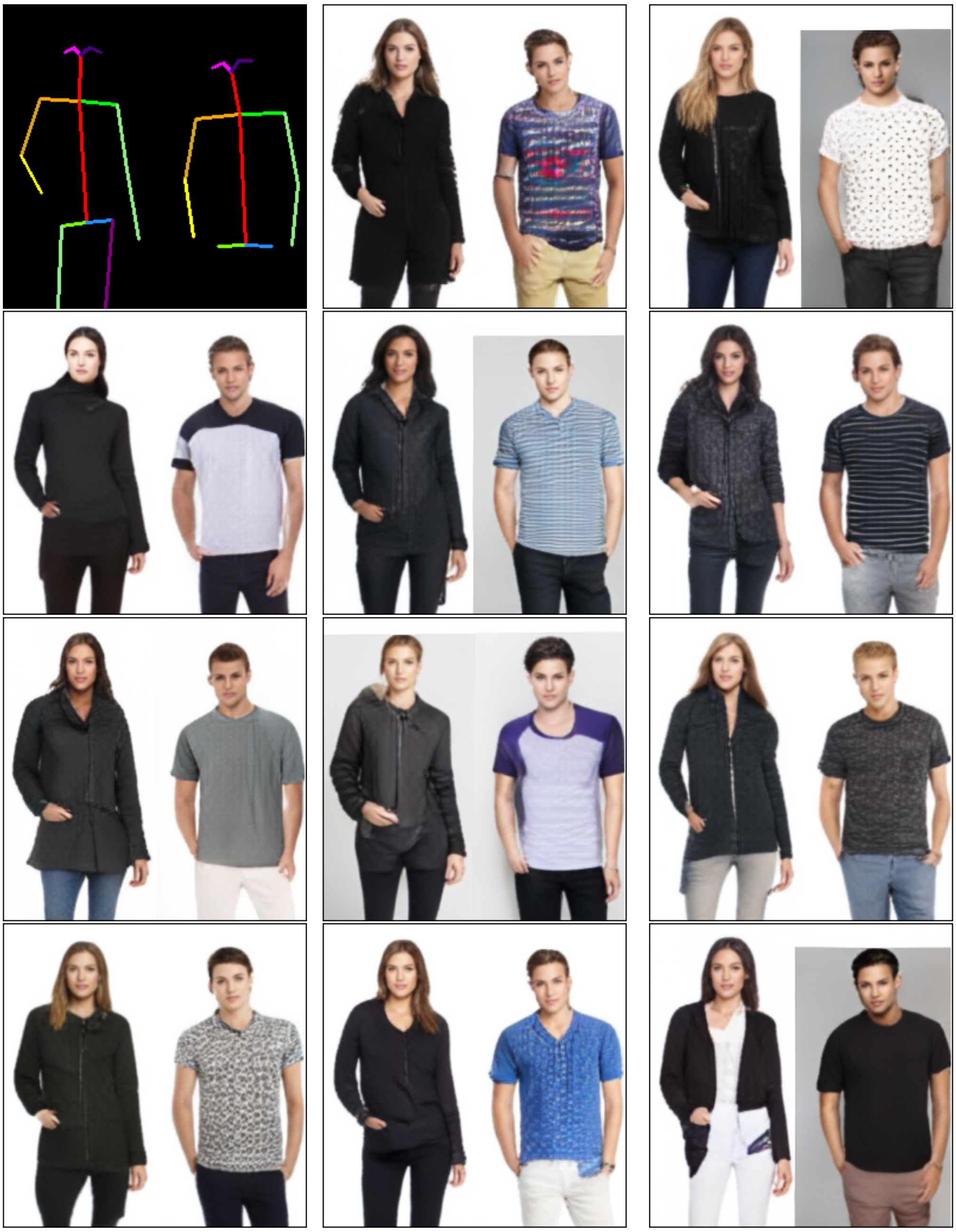


Figure 3.8: 'the lady is wearing a black long-sleeved parka.the man is wearing a multi-colour short-sleeved tee.'

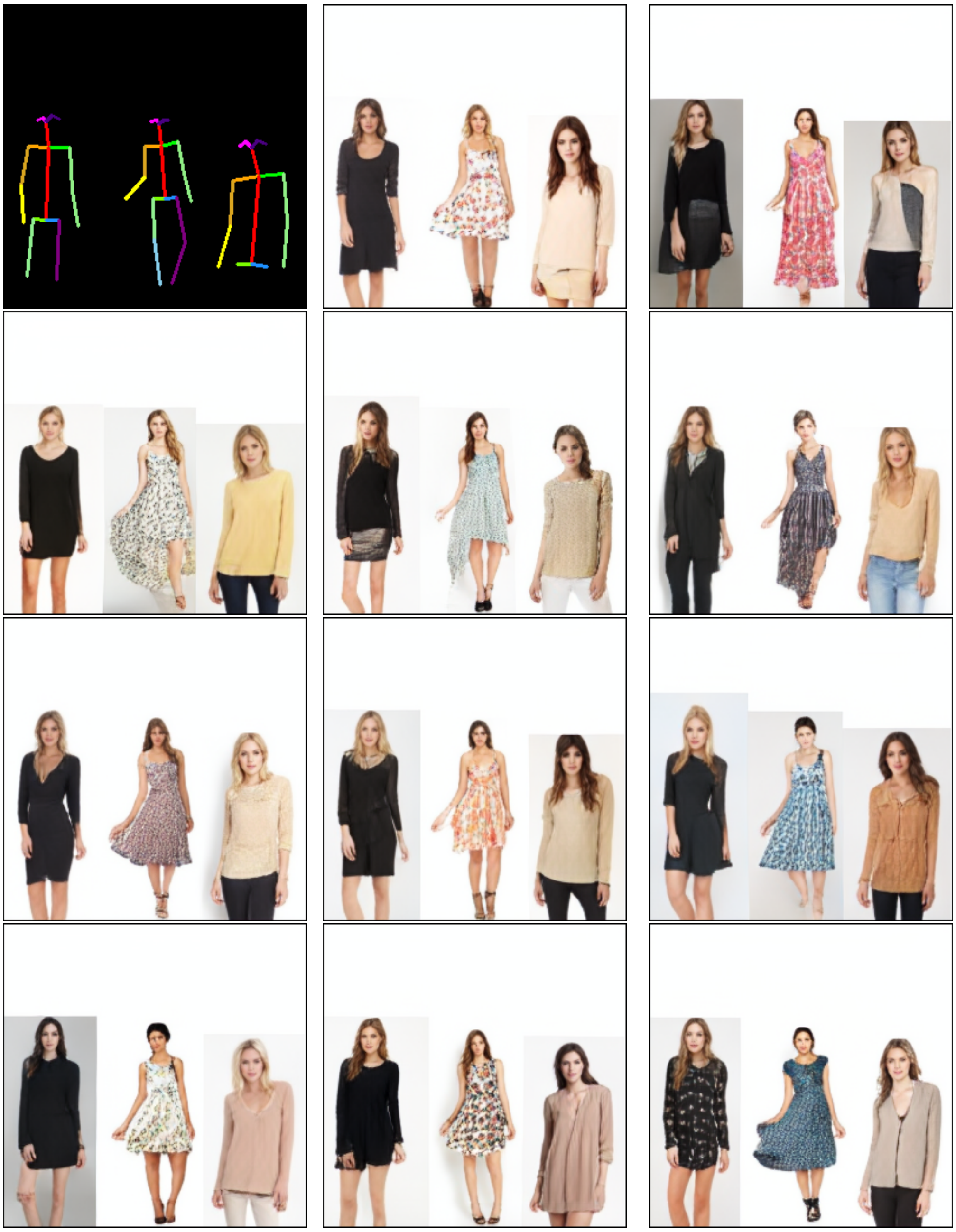


Figure 3.9: ‘the lady wears a black long-sleeved romper.the lady wore a multicolour sleeveless dress.the lady is wearing a blouse with a long sleeved khaki.’

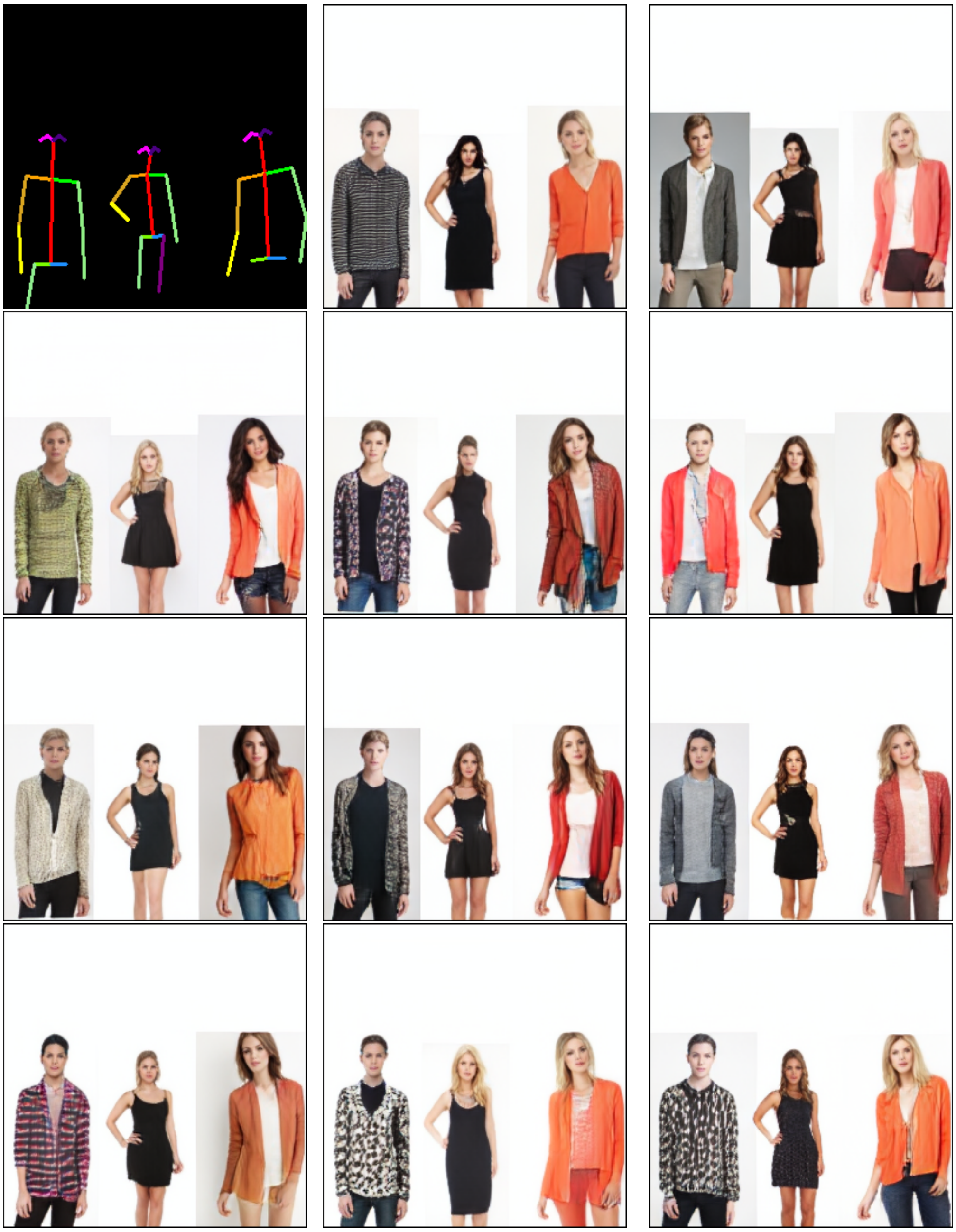
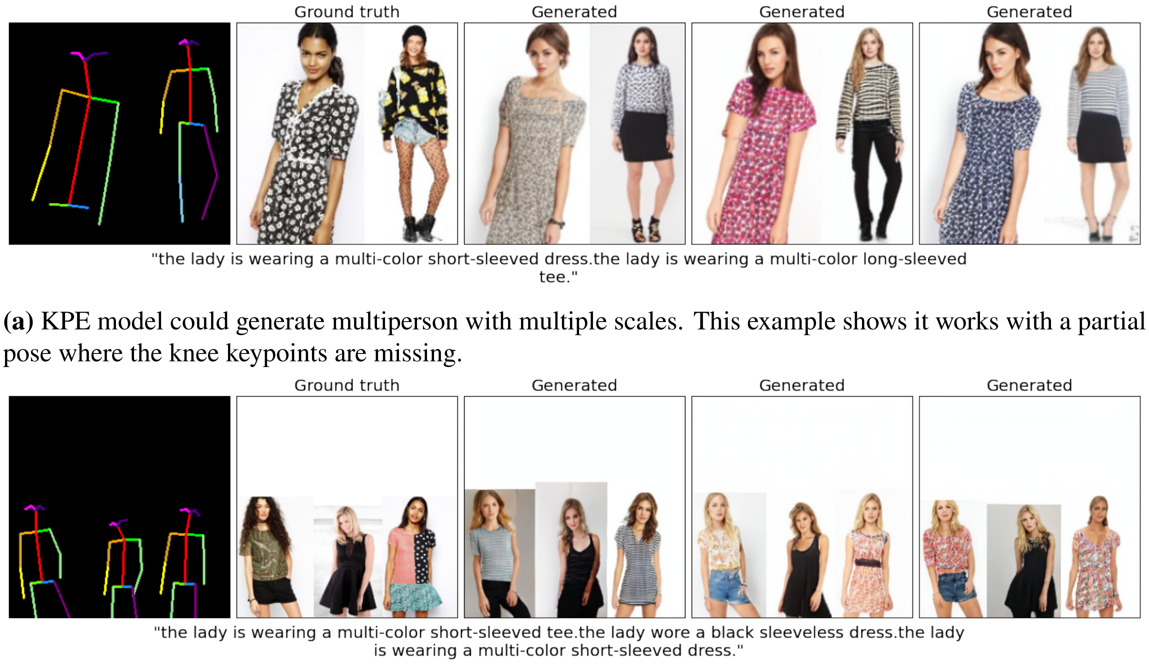


Figure 3.10: 'the man wore a cardigan with a multicolour long sleeve.the lady wore a black sleeveless dress.the lady wore a long orange blazer.'

Pose Method	DALL-E	DALL-E+VQGAN	KPE (Ours)
Number of pose tokens ↓	-	256	**25**
Relative inference speed ↑	**1.73×**	1.0×	**1.73×**
FID ↓	22.11	21.81	**20.39**
PCE ($\times 10^{-3}$) ↓	8.2	1.2	**0.6**
CLIPSIM ↑	**0.27**	**0.27**	**0.27**
IS ↑	2.912	3.027	**3.034**
OKS ↑	0.598	**0.970**	**0.970**
Mask-SSIM ↑	0.265	0.420	**0.424**

Table 3.1: Evaluation of different models on DeepFashion multiperson dataset. Our method, KPE, achieves the highest scores in all metrics.

For quantitative results, Table 3.1 shows that our proposed method, KPE tops all the evaluation metrics; it achieves the highest FID and IS scores, indicating that KPE can generate realistic looking people. Figure 3.11 and 3.1(a) show examples with various genders and quantity of people, with different scales, poses and occluded poses with missing keypoints. Given the OKS score of 0.97, which indicates a highly accurate pose, the high Mask-SSIM score suggests the generated images have gender and clothing appearance matching the text description. The CLIPSIM is the same for all methods despite DALL-E and DALL-E+VQGAN having worse PCE. This suggests CLIP [86] trained on general images is not good at spotting human body errors.



"the lady is wearing a multi-color short-sleeved dress.the lady is wearing a multi-color long-sleeved tee."

(a) KPE model could generate multiperson with multiple scales. This example shows it works with a partial pose where the knee keypoints are missing.

"the lady is wearing a multi-color short-sleeved tee.the lady wore a black sleeveless dress.the lady is wearing a multi-color short-sleeved dress."

(b) A variety of different poses by three people. Each of the people matches the text description in gender and clothing appearance.

Figure 3.11: KPE model can generate photorealistic people with an accurate pose. This figure shows the pose illustration, ground truth, and three generated samples. The ground truth images are not used in the inference, they are included only for comparison.

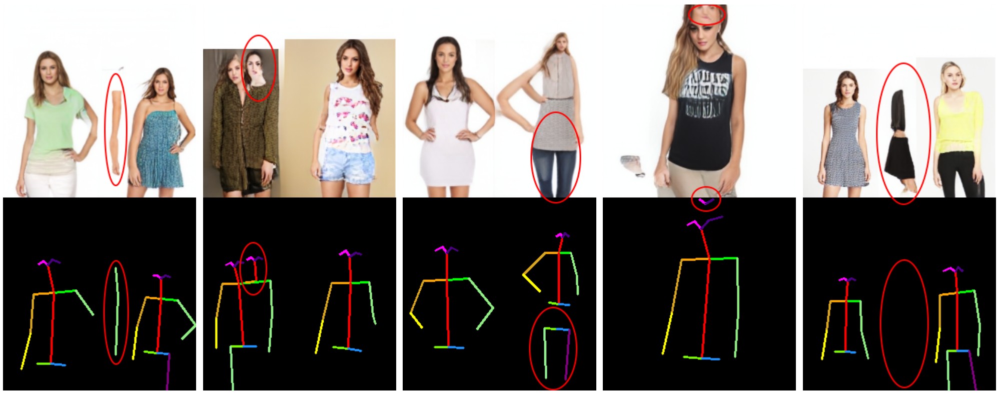
3.5.2 Comparison with DALL-E+VQGAN

Both KPE and DALL-E+VQGAN produce high-quality people images with an accurate poses. From Table 3.1 against the baselines DALL-E and DALL-E+VQGAN, we can see that its OKS score matches KPE and is only marginally behind in IS and mask-SSIM, but PCE is twice the error rate of KPE. Apart from generating improved images, there are several advantages of using KPE that make it an overall superior method:

- **Less memory**. The keypoint token length is smaller than the pose image token, requiring less computational memory. In our experiment, the image token length is 256 while there are only 25 keypoint tokens, making it at least 10× more memory efficient.
- **Faster to run.** The reduction of token number reduces computational complexity, which is in $O(N^2)$ for the transformer. Due to limitations in profiling tools, we measured end-to-end inference time without a detailed breakdown of time spent on individual stages such as image encoding and autoregressive sampling. Our KPE model achieves faster inference than DALL-E+VQGAN, requiring only 58% of the baseline inference time on an RTX5000 GPU. Notably, despite the inclusion of additional pose tokens, our method showed no significant speed difference compared to the baseline DALL-E. This suggests that image encoding with VQ-VAE is a major computational bottleneck, and eliminating this step could lead to substantial speed improvements.
- **Invariant pose representation.** The same VQ-VAE encodes both the pose and target images. However, as VQ-VAE is normally pretrained on natural images, thus the trained VQ-VAE may not perform well on synthetic skeleton images. In contrast, KPE relies only on the keypoint information and is invariant to the image nor VQ-VAE.
- **Scalable.** Increasing target image resolution or quality will require an increase in image token length hence more memory and slower running. Since KPE is invariant to the image, the pose processing will not increase computational resources as the image resolution increases. This makes our method easier to scale to higher image resolution.

3.5.3 Ablation Study and PCE

We did an ablation study comparing KPE against the baseline [90], a text-to-image model without pose guidance. KPE tops all the metrics in Table 3.1, most notably with PCE rate at 0.6 $\times 10^{-3}$, which is over 13 times better than baseline's 8.2 $\times 10^{-3}$. Figure 3.12 shows an example of images that contains errors and how we can spot the error by using PCE. Figure 3.12a contains two realistically looking people but with an additional long arm floating in the centre of the image. The floating arm is assigned to the third person, causing PCE=1 as $h(x) \neq gt(x)$. Although measuring the discrepancy in people's count does not catch every error, it allows a standardised metric across a wide range of examples without the need for a manual visual inspection of each. Figure 3.12b to 3.12d show examples of additional body parts where PCE=1, some of which can be difficult to see initially, like the additional face in Figure 3.12d. Also, we found that the baseline [90] sometimes generates fewer or more people than the text description. When it happens, it tends to contain some standalone body parts like Figure 3.12a and Figure 3.12e. The PCE



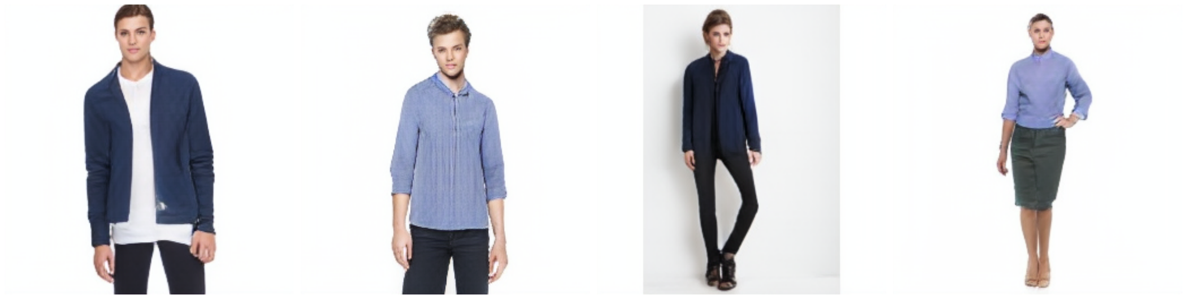
(a) gt=2, h=3 **(b)** gt=2, h=3 **(c)** gt=2, h=3 **(d)** gt=1, h=2 **(e)** gt=3, h=2

Figure 3.12: Top row are erroneous images generated using DALL-E, and the bottom row shows the keypoints obtained from the images. PCE can capture image errors by comparing the generated (gt) and intended (h) number of people.

can pick up the error in Figure 3.12e despite OpenPose failing to detect the incomplete person in the centre.

3.6 Limitations

The DeepFashion dataset is hugely imbalanced, where men form only a tiny fraction of the dataset, and the rest are long-haired white females. Therefore, a pose-only guided model trained on the dataset is more likely to generate female images. To understand the effect of the gender bias, we generated images of various poses using the exact text prompt "a man wore blue shirt" as shown in Figure 3.13.



(a) **(b)** **(c)** **(d)**

Figure 3.13: Text of "a man wore blue shirt" was used to generate images from pose that is more masculine (a) towards more feminine pose in (d).

We can see in Figure 3.13a and 3.13b that despite the gender bias in the dataset, our model can generate convincing men from neutral poses. However, when presented with poses that are exclusive to females in the dataset, the generated images (Figure 3.13c and 3.13d) lean toward feminine appearance. This limitation implies that pose is not entirely disentangled from gender, and the model learned the gender bias from poses containing information about body proportion. Given this insight, in future, we will collect and apply our approach to a more balanced dataset in the future to address this bias in gender and ethnicity.

Figure 3.14 shows how early result of applying our method to generate image sequence by controlling consistency of person appearance and pose. However, text conditioning alone is not strong enough to for precise control of the fine details such as clothing length. This leads us to applying visual conditioning in next chapter for better conditioning.

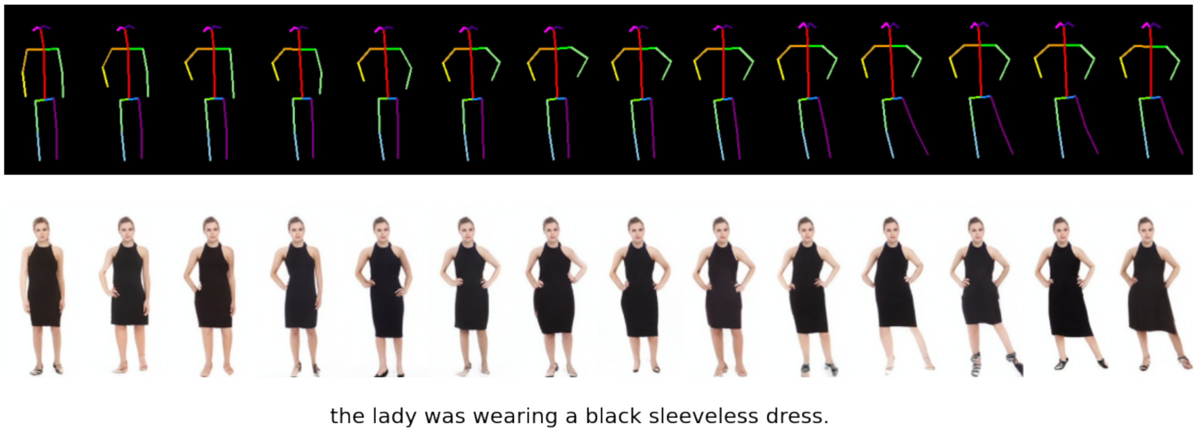


Figure 3.14: A series of images generated by conditioning on the text and sequence of poses. The top rows of image tokens covering the head are also fixed as conditions to generate subsequent image tokens.

3.7 Conclusions

We have introduced a combined approach using text and pose keypoints as guidance in autoregressive transformer, effectively producing high-quality, photorealistic multi-person images. Our results demonstrate that adding pose information significantly enhances image quality beyond the capabilities of state-of-the-art text-only models. The Keypoint Pose Encoding (KPE) approach also matches or surpasses performance compared to image-token-based methods, with a substantial reduction in computational demand. Additionally, we introduced a new metric, People Count Error (PCE), designed to accurately detect errors in the representation of human figures in generated images, thereby providing a more comprehensive assessment method for model performance in human image generation.

While our pose conditioning is effective, we also learned the limitations of the transformer architecture. The insufficient in long range attention for among image tokens can lead to additional or wrong body parts being generated. This led us to explore new model architecture in the next chapter.

Chapter 4

Fine-grained Visual and 3D Pose Control for Diffusion Models

In last chapter, we successfully applied parametric 2D keypoint tokens to autoregressive transformer for pose conditioning. However, the underlying transformer and attention mechanism caused incorrect body parts to be generated. Furthermore, text conditioning alone is ambiguous to enforce image consistency which is essential for many real life applications. Therefore, in chapter, we explore using a different model architecture - diffusion model. We explore a novel method of pose conditioning using parametric 3D pose data, alongside fine-grained visual conditioning.

Code is available from `https://github.com/soon-yau/upgpt/`

4.1 Introduction

Generating humans from text and/or pose is a challenging problem in computer vision. The methods can be classified into two categories, (1) image generation (synthesis) and (2) pose transfer and editing. Image generation can be unconditional or conditioned on other information *e.g.*, pose and text. Pose-guided image generation, conditions on pose (keypoints, skeleton image, heatmap, body mesh) to generate images [1, 26, 48, 81]; and text-to-image models such as DALL-E[89, 90] and [99, 113, 131, 138, 139, 153]. Pose or text to image is a one-to-many mapping. It can create a person with vastly different appearances even given the same conditions - *e.g.*, a person in the same pose but wearing other clothing or shades of colour from the word “red shirt.” The ambiguity and inconsistency prohibit them from being used to perform image editing, which requires the maintenance of the visual appearance of all other aspects of the image apart from the elements or regions to be edited. Some newer methods [16, 51, 142] use pose and text to exert further control. However, the effect is still limited by the inherent ambiguity of these modalities and is hence unsuitable for image editing.

The other category is image editing, for tasks such as changing the clothing, human pose, or face. Most pose-guided image generation literature fall into this category, performing pose transfer to transfer a person’s appearance from a source image to the pose of a target image. However, we prefer the term *edit* to encompass other forms of modification, including using text or modifying the pose parameters directly, rather than having to *transfer* them from the other image. Pose transfer models [72, 105, 133] use both human pose and a source image as conditions for the generative image model where visual information of a source image serves as a stable condition to encourage the models to maintain a person’s appearance in the generated image. [91, 140, 150, 155] have extended capabilities that could also transfer

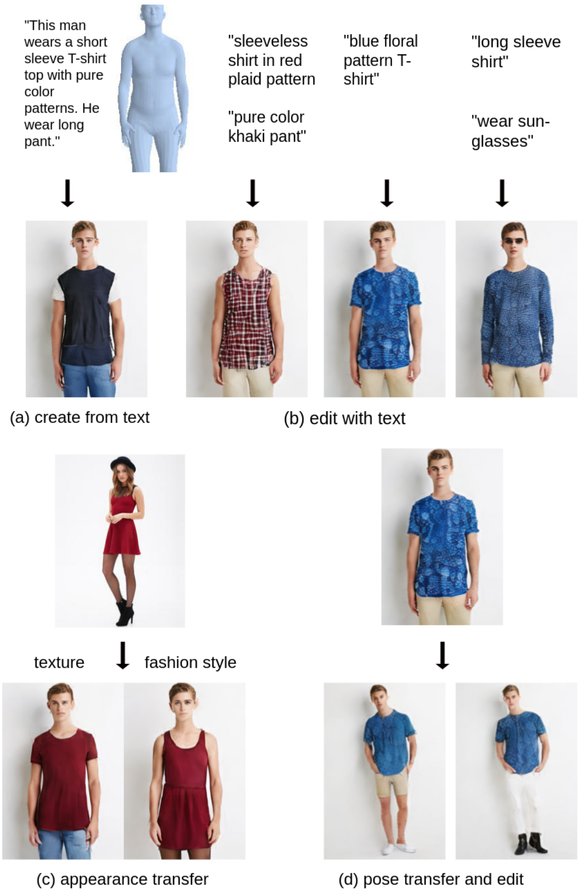

"This man wears a short sleeve T-shirt top with pure color patterns. He wear long pant."
"sleeveless shirt in red plaid pattern
"pure color khaki pant"
"blue floral pattern T-shirt"
"long sleeve shirt"
"wear sun-glasses"
(a) create from text
(b) edit with text
texture
fashion style
(c) appearance transfer
(d) pose transfer and edit

Figure 4.1: UPGPT can perform all person image generative tasks: (a) text and pose guided image generation, (b) fine-grained, mask-less region editing with text, (c) style and appearance transfer, (d) pose transfer followed by edit.

texture, clothing shape, or both, *i.e.*, appearance transfer, but no single model can perform all those tasks. More importantly, they all need to train on source-target image pairs and can not generate a new image without a source image. To bridge the gap between the two categories, we propose *UPGPT* to perform both generation and edit tasks using a single trained universal model, and image sampling pipeline, as seen in Figure 4.1.

In our research, we discovered four underlying problems in person image generation and editing that have yet to be addressed: **(1)** Existing methods cannot interpolate human pose due to the inherent limitation of the chosen pose representations. 2D body segmentation map (parsing map) and body mesh are dense representations (pixel and voxel) and cannot be interpolated. To interpolate 2D keypoint points and their derivatives (skeleton image, heatmap), they must first be mapped into 3D space, which is a difficult task on its own, before performing the interpolation in 3D space, then project back into 2D keypoints. We break away from the tradition by using pose parameters of SMPL[69], a 3D body model that represents pose by rotation of body joints. Then, performing linear interpolation on the SMPL parameters produces pose interpolation using our model. **(2)** Existing person image editing methods require parsing maps, which is difficult for users to create or edit by hand. Furthermore, their methods are typically constrained to transferring information from a single modality. To address this challenge, our method allows text or drag-and-drop of the reference image or a combination of them to perform convenient and fast image editing. **(3)** Missing information from the source image. For example, when a target image expects a full-body person, the source image only contains a partial view where the lower part is not visible, as shown in Figure 4.2. This leaves a question of whether the model should generate short pants, long pants, a dress, a sneaker, high heels, or leather shoes. **(4)** A person's appearance can change in the target image *e.g.*, a person wearing a jacket in the source image may have it taken off in the target. Existing methods rely solely on the source image to provide all the information. Still, they can fail to generate desired or correct results if the information is incomplete or wrong, as shown in Figure 4.7. We address problems 3 and 4 by adding a new modality - text to enrich the information source and to reduce and correct errors. The text description of the expected outcome can work as a way to filter out unwanted information (not wearing a jacket) or to fill in missing information (to generate pants or skirts).

Table 4.1 compares the capabilities of the two main person image generation methods, and our proposed method combines all the key features. In summary, the main contributions in this chapter are:

1. A unified framework that can simultaneously perform person image generation, editing, and pose transfer tasks.
2. The provision of zero-shot, mask-less image generation and editing with text.
3. The use of 3D parametric body model parameters to demonstrate the first simultaneous pose and camera view interpolation.

	Pose Transfer	Text-Pose-to Person Image	UPGPT (Ours)
Pose Edit	✓	✗	✓
Appearance Edit	✓	✗	✓
Texture Edit	✓	✗	✓
Create from Text	✗	✓	✓
Edit with Text	✗	✓	✓
Pose Interpolation	✗	✗	✓

Table 4.1: Comparing the superset capabilities of pose transfer[74, 91, 92, 133, 140, 150, 155], text-pose-to-image [16, 51, 142] and our method. Our unified method can perform all the person generation and edit tasks and introduce a new capability of pose interpolation.

4.2 Related Works

Diffusion Models [22, 40] have shown superior image quality and text-guided capability. [2, 3, 79, 89, 99] enable image editing by performing text-guided diffusion on regions defined by segmentation mask. Dreambooth[97] shows that they could encode a person's face into a text token and use a diffusion models to generate the person in a different scene. Prompt-to-Prompt[36] proposed a mask-less edit of coarse objects by learning the region from the attention map. Our method achieves mask-less editing by learning and disentangling a person's appearance.

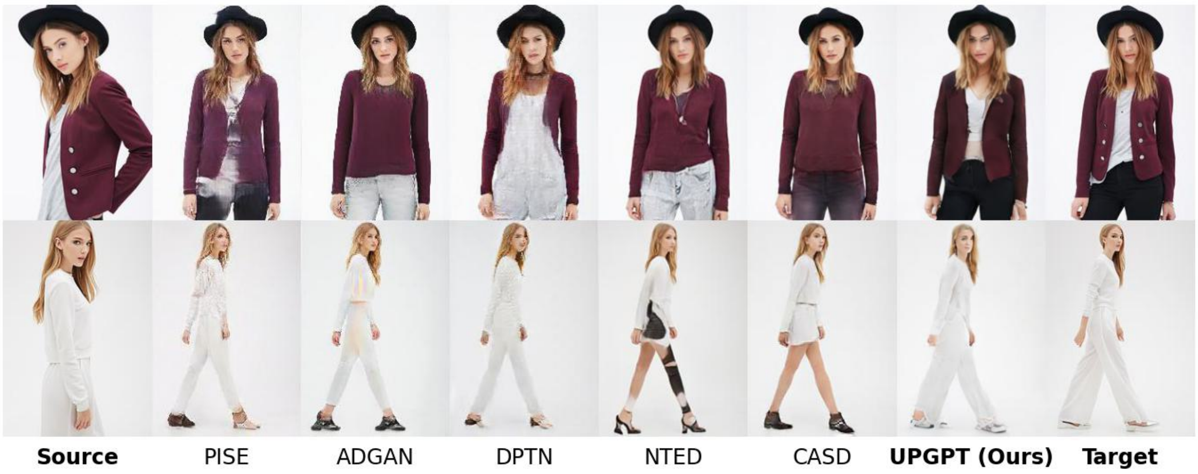
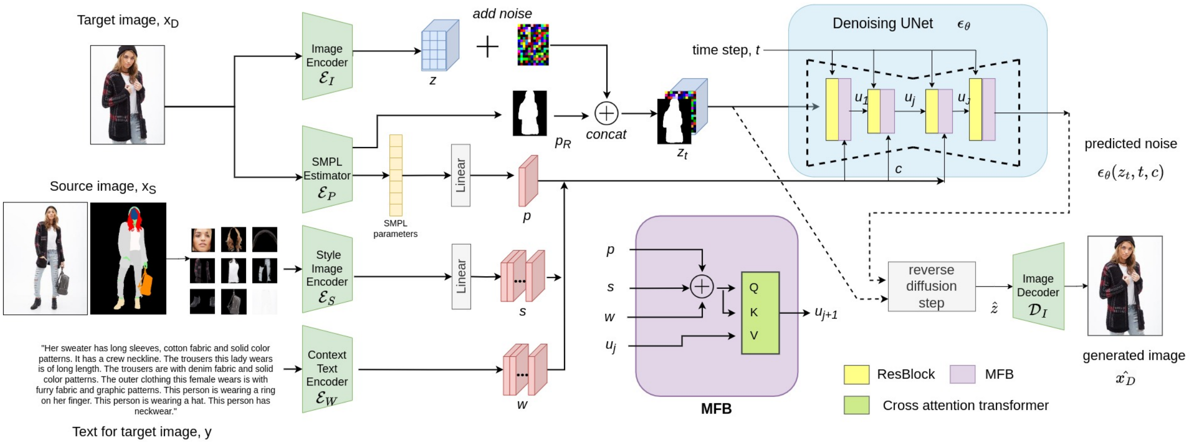


Figure 4.2: Pose transfer is an ill-posed problem: Often, the source image does not contain all information for the target pose *i.e*., in this figure, the pant. Compared to existing methods (PISE[140], ADGAN[74], DPTN[144], NTED[91], CASD[150]), our method can create the desired result by utilising additional multimodal information.

Pose Guided Image Generation. Ma *et al*. [72] was among the first literature on pose transfer; they concatenated source images with the target pose heatmap and used them as input conditions to a GAN[31]. Starting from PATN[155], models take pose from both the source and image. Yang *et al*. [133] detected and cropped out the person's face and used that as an additional image condition within the network for a more fine-grained detailed generation. In addition to human pose, ADGAN [74] uses a human parsing map to segment the body parts of the source image to extract their style codes. This allowed them to change the style or texture of clothing region. However, as the shape of the person and clothing is bounded by the segmentation map of the source image, the image edit is limited to only texture transfer. To overcome this issue, PISE [140] and SPGNET [71] trained a separate network to generate a parsing map of the target pose, which they edited before feeding into the image generator. Allowing the changing of clothing shape *e.g*., from short sleeve to long sleeve, but they cannot perform texture transfer simultaneously. While NTED[91] and CASD[150] demonstrated transfer of the entire clothing pieces, they do not provide a method to transfer only the texture. DPTN[144] uses two paths - source-to-source and source-to-target, while we require only one path for both trainings. Unlike our approach, existing methods can only edit a subset of clothing texture, shape, or appearance (texture and body), but not all of them. [1] uses SMPL body mesh, which is more computationally expensive to process than our method, which uses only 72 parameters. Concurrent to our work, PIDM[7] shows clothing style interpolation by interpolating the diffusion models's noises, but they could not perform pose interpolation.

Text-Guided Image Generation. There exist text-to-image models since the early days of GANs [131, 139, 153], to transformer[120]-based DALL-E[90] and diffusion models[79, 89, 95, 99]. However, they do not provide precise control over human pose or fine-grained appearances. KPE[16] created the first text-and-pose-guided image generative model that encodes body keypoints into transformer tokens as conditions. Although it can generate accurate poses, as text is a weak condition, it cannot provide fine-grained appearance control and consistency for pose transfer. Using a parsing map and hierarchical autoencoder to encode different body regions, Text2Human[51] offered more fine-grained appearance control. Still, it could not specify person and clothing attributes not labelled in the text description, notably the clothing colour. HumanDiffusion[142] segments and encodes each clothing item with CLIP image encoder into style code and uses a fixed-size database to store the embedding of fashion styles. During sampling, they use either CLIP image or text embedding, but not both, to retrieve the closest embedding from the database. Although this allows them to control the clothing colour using text, their method entangles the clothing type, colour, and texture pattern into a finite number of combinations. In contrast, our method offers disentanglement and allows users to edit each clothing attribute independently using combination of image and text. The existing generative methods could not consistently generate images for pose transfer or appearance editing. More recently, ControlNet [143] adds pose guidance to diffusion models, but it cannot ensure appearance consistency due to the lack of visual conditioning.

4.3 Methodology



"Her sweater has long sleeves, cotton fabric and solid color patterns. It has a crew neckline. The trousers this lady wears is of long length. The trousers are with denim fabric and solid color patterns. The outer clothing this female wears is with furry fabric and graphic patterns. This person is wearing a ring on her finger. This person is wearing a hat. This person has neckwear."

Figure 4.3: Overview of our proposed UPGPT architecture. In training, we encode pose, style image, and context text into embeddings that go to the Multimodal Fusing Block (MFB) for fusing. The output of MFB is used as a condition in UNet to predict the noise needed to denoise the image's latent. In sampling, the image encoder decodes the denoised latent $\hat{z}$ into pixel space.

The primary motivation of our proposed method is to fully disentangle a person's image into content and style represented by pose, text, and image features. We can independently edit and mix the different modalities at source to provide fine-grained person image generation and editing. Figure 4.3 illustrates the overall architecture of UPGPT. The first step is to extract the person's information from images and text in the form of features and encode them into conditioning embeddings. The second step is to fuse the embeddings within Multimodal Fusion Block (MFB) to provide conditioning to the UNet of the diffusion models. The figure shows the training pipeline for the pose transfer task with the source-target image pair at the input. However, this can be repurposed for image generation tasks using the same image as the source and target image. Existing person image generation methods [16, 51, 142] use only individual images in training, while image pairs are necessary for pose transfer methods [7, 74, 91, 92, 133, 140, 150, 155]. Our novel architecture allows us to use individual and paired images to increase the training sample size. The following section describes the proposed method in detail.

4.3.1 Multimodal Feature Representation

Our model uses three modalities: pose, image, and text. We further divide the text into context text and style text. Overall, a person's image is disentangled into content represented by pose and context text;

and style as defined by style text and image.

Image Latent. We encode the target image $x_D \in \mathbb{R}^{H\times W\times 3}$ using VAE's [58] encoder into the latent variables $\mathcal{E}_{\mathcal{I}}(x_D) = z \in \mathbb{R}^{\frac{H}{f},\frac{W}{f},d_V}$ where d_V is the VAE's channel dimension, and f is a downsampling factor in the power of two, and x_D and z are only needed in training. In image sampling, the trained diffusion models generates a new image latent $\hat{z}$, to be decoded by the VAE decoder into pixel space $\hat{x_D} = \mathcal{D}_{\mathcal{I}}(\hat{z})$. Smaller f *e.g.*, 4 gives higher spatial resolution but quadruples the latent size from $f = 8$ and thus increases computational effort considerably. Although large f is more computationally frugal, the resulting z has a smaller spatial dimension, which will store more visual details for the same pixel patch. As a result, a small face in a full-body image can appear blurry after image reconstruction $\mathbb{D}_I(\mathcal{E}_{\mathcal{I}}(x_D))$.

SMPL Pose. We use [141] as pose estimator $\mathcal{E}_{\mathcal{P}}$ to create an embedding based on the SMPL parameters from the target image x_D. The 72 SMPL parameters represent three axis-angle rotations of 24 body joints, ten body shape parameters, and three camera parameters. The camera view of an image is determined by the body's vertical axis rotation parameter and the camera parameters. Each of the three camera parameters in Cartesian coordinate axes determines horizontal translation, vertical translation, and zooming. The SMPL parameters are flattened and projected with a linear layer to $p \in \mathbb{R}^{1\times d}$ where d is the context text embedding channel dimension. Experiments show that the SMPL's camera parameters are insufficient to ensure the person's correct horizontal position. Therefore, we concatenate a silhouette mask $p_R \in \mathbb{R}^{\frac{H}{f},\frac{W}{f}}$ at the UNet input to reinforce the pose conditioning, we call it as **reinforced person mask (RPM)**. RPM only needs to be a coarse mask; this differs from [71, 74, 140, 150], which requires a detailed body part segmentation map. We used binary silhouette mask in our main experiments but tried other methods as discussed further in Section 4.4.8.

Style Image. From a source image x_S, we use a segmentation map to segment the person into 9 fine-grained semantic regions *i.e.*, head, hair, headwear, background, top, bottom, outwear, shoes, and bag. Each of the segmented regions is cropped and resized. We call this style image, and we use it as a condition for the person's appearance style. Unlike conventional methods that perform segmentation in run time, we do it in the data preparation stage and store the style regions. This provides image editing flexibility by simply changing the style image files. We do not use source images any more after obtaining the style images. We treat a person's identity as one of the styles determined by face and hairstyle images. We use a separate face detector to normalize the face - align the face to an upright position. If an occluded face is not detected, we replace it with another normalized face image from the same person if it is available. We encode the style images with a pre-trained CLIP [86] image encoder $\mathcal{E}_S$ before projecting it with a linear layer into $s \in \mathbb{R}^{N\times d}$ where N is the number of style regions defined for a person.

Style Text. CLIP [86] trains an image encoder and text encoder jointly on image-text pairs, with a common embedding for both modes aiming to be close to each other in the CLIP embedding space. For example, the CLIP embedding of the text "a red shirt" and an image of a red shirt should be close in terms of Euclidean distance. We use this to create a zero-shot learning method through editing with text. Like us, HumanDiffusion[142] uses CLIP image encoding in training, but they can only use either text or image to control image sampling, while we can use either or both modalities. Also, we use two different text conditions - content and style to provide better disentanglement and finer control. Figure

4.4 shows how we can mix the style images and texts in image sampling. Style text provides a fast and convenient way to control the clothing texture and colour, while we can use style images to dictate specific appearances such as face and colour shade.

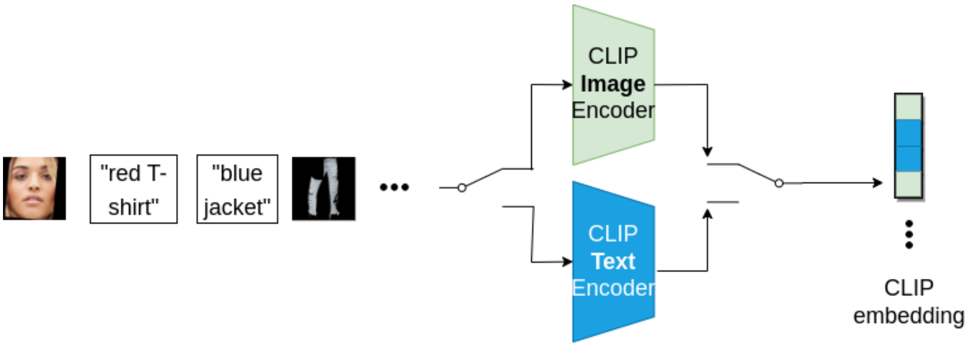


Figure 4.4: We can mix-and-match a combination of image (green) and text (blue) embedding in sampling time.

Content Text. The content text describes the content of the target image *e.g.*, gender, clothing shapes, and fabrics. We use a pre-trained LLM (large language model) transformer [47] for text encoding. We take the transformer's last layer feature as our content text embedding $\mathcal{E}_{\mathcal{W}}(y) = w \in \mathbb{R}^{l \times d}$ where l is the maximum text token length.

4.3.2 Conditional Diffusion Model

The diffusion models training process consists of a sequence of time steps $t = 1...T$, where Gaussian noise ϵ is scaled using a noise schedule [40] and added to an image latent variable z to produce a noisy version. This concatenates with p_R to produce z_t, fed into the input of a denoising UNet ϵ_θ. We propose to condition using our MFB block to concatenate $\oplus$ pose p, text w and style embedding s and perform cross-attention with UNet's ResBlock output at every level u_j where j is layer number.

$$c = p \oplus s \oplus w, Q = \phi_Q(c), K = \phi_k(c), V = \phi_V(u_j) \tag{4.1}$$

where ϕ performs 1×1 convolution layers for projection into u_j's channel dimension d_j and flatten to 1-dimension.

$$CrossAtten(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_j}})V \tag{4.2}$$

We train the UNet by using MSE loss on predicted noise $\epsilon_\theta(z_t, t, c)$:

$$\mathcal{L}_{MSE} := \mathbb{E}_{z, p_R, c, t, \epsilon \sim \mathcal{N}(0,1)} \left[\| \mathcal{W} \odot (\epsilon - \epsilon_\theta(z_t, t, c) \|_2^2) \right] \tag{4.3}$$

Where $\odot$ is element-wise multiplication and $\mathcal{W} \in \mathbb{R}^{\frac{H}{f}, \frac{W}{f}}$ is loss weight we add to the standard diffusion loss. In addition to the primary loss, many GANs[91, 92, 140, 150] use perceptual loss[52], which extract features from image pixels. However, a single training step in the diffusion models does not generate an image; therefore, we cannot directly use additional losses that require image pixels. Consequently, we use a loss weight $\mathcal{W}$, a 2D tensor with the same dimension as the image latent, to assign different weights to the loss. This helps to regulate the training under challenging regions such as face and hands.

4.3.3 Generation, Transfer & Editing of Images

Unlike previous pose transfer work, we do not need to use a segmentation map or any reference person image when sampling a new image. We create a new random image latent z_0 to begin the sampling process. Progressively in each time step t, the image latent is denoised using the reverse diffusion step as described by [40] to produce a less noisy image latent $\hat{z_t} = G(z_t, t, c)$. After the T steps, the denoised $\hat{z}$ is decoded by the VAE decoder $\mathcal{D}_{\mathcal{I}}(\hat{z})$ to create an image in pixel space. We use the same pipeline for all the tasks by changing only the conditioning.

To adjust the clothing texture and colour, we can either do a texture transfer by using a style image or by replacing the style embedding for that clothing with style text, all without a segmentation mask. Due to the suitable disentanglement property of our method, this changes only the texture and colour but not the clothing shape, as demonstrated in the left image in Figure 4.1(c). If we fix the style condition and change only the context text *e.g.*, from "long sleeve" to "short sleeve," it will only change the sleeve length while maintaining the clothing texture. We can modify the content text and style for appearance edit/transfer, which change/copies both the shape and styles. To perform pose transfer, we replace the pose of the source image with one from the target image; this would produce results similar to existing pose transfer methods. On top of that, we use context text from the target image that better describes the desired appearance to generate images with clothing appearance more faithful to the target image. The different configurations are summarized in Table 4.2, and some image examples are shown in Figure 4.1.

Task\Condition	**Styles**	**Content Text**	**Pose**
Generate	source	source	source
Texture Edit	style image/	source	source
Shape Edit	source	target/edit	source
Appearance Edit	style image/	target/edit	source
Pose Transfer	source	target	target

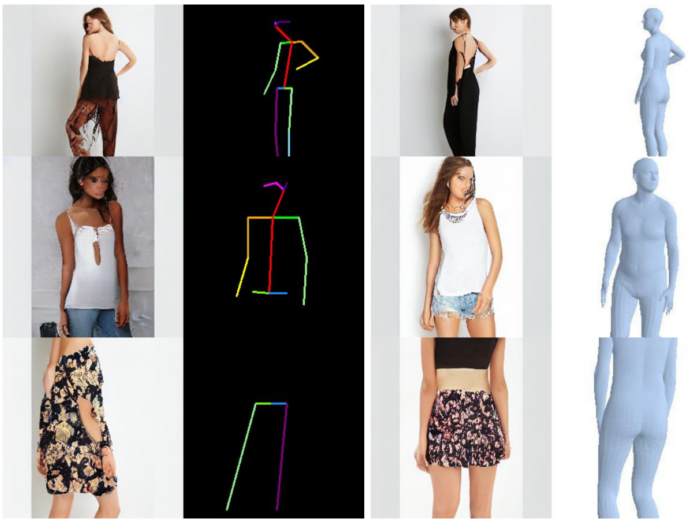
Table 4.2: Starting from image generation using information from the source image, the table shows how our method can perform various tasks using different conditioning combinations.

4.4 Experiments

In our preliminary experiments, we conducted a small-scale comparison of Keypoint Pose Encoding (KPE) with SMPL-based conditioning and found that SMPL conditioning outperformed 2D keypoint encoding when the keypoints provided were sparse, as illustrated in Figure 4.5. This suggests that SMPL offers more robust pose control under limited input data conditions.

Then, we performed large scale experiments on two tasks: (1) text-pose guided image generation and (2) pose transfer. Both use the same model architecture but different image resolutions and subsets of the DeepFashion dataset [156].

Implementation Details. We train our model using AdamW optimizer [70] at a learning rate of $5{\times}10^{-5}$,



(a) KPE (b) SMPL

Figure 4.5: Comparison between pose conditioning with KPE and SMPL. SMPL has similar performance with KPE when sufficient number of keypoints are provided, but outperform KPE when only sparse keypoints is present in the last row.

batch size of 24, and loss weight, $\mathcal{W}$ (Equation 4.3) used is face=8.0, arms=2.0, background=0.5 and 1.0 for others, and a silhouette mask is used for reinforced pose mask. Our model is trained with $T = 1000$ noising steps and a linear noise schedule.

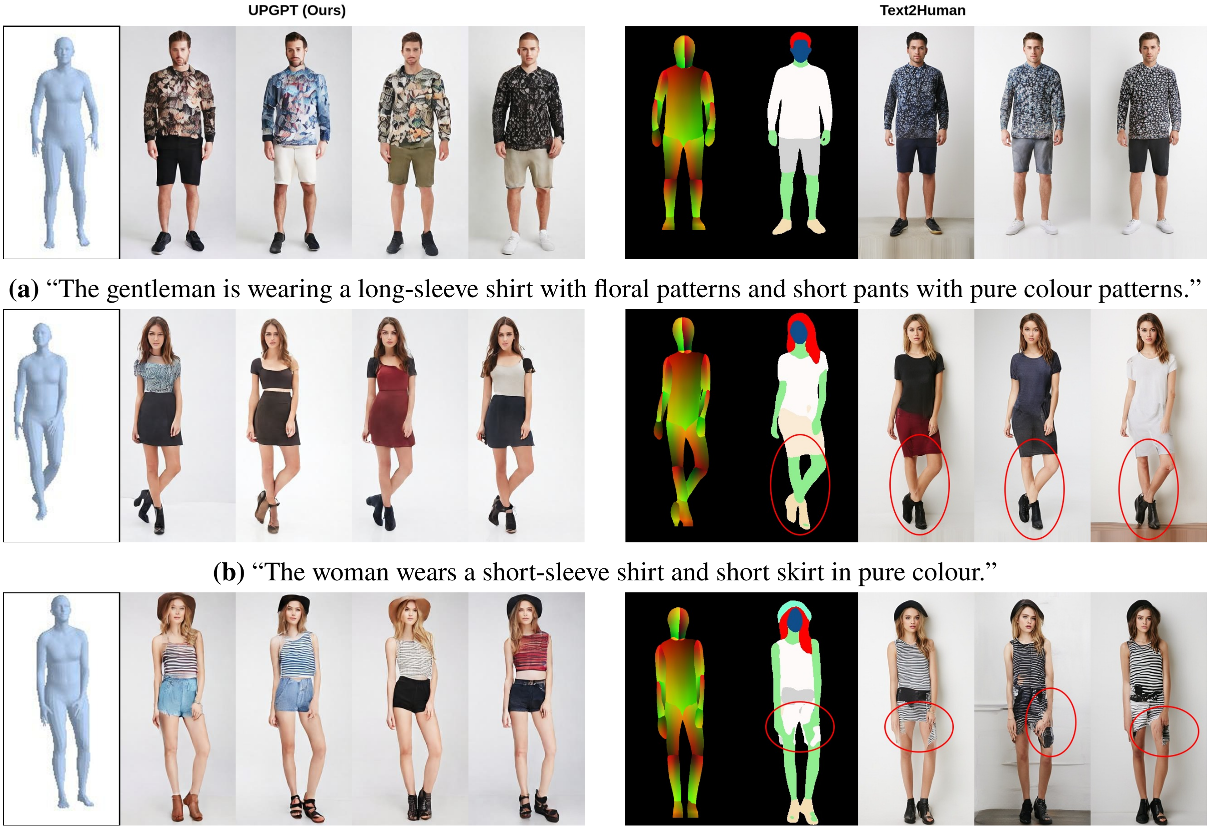
Evaluation Metrics. We use *LPIPS*[145] and *SSIM* [125] to measure the similarity between the generated image and target image in the pose transfer task. LPIPS uses pre-trained VGG[106] to calculate the perceptual similarity, while SSIM measures the similarity by considering the images' luminance, contrast, and structure. For the text-pose guided image generation task, clothing colour changes can significantly impact the similarity score, even if it looks realistic. Therefore, instead of comparing individual images, we use Frechét Inception Distance (*FID*)[37] to measure the distribution of two groups - ground truth and generated images.

4.4.1 Text-Pose Guided Image Generation

Method	**FID↓**
†HumanDiffusion[142]	30.42
Text2Human[51]	24.52
UPGPT(Ours)	**23.46**

Table 4.3: Quantitative result on DeepFashion Multimodal dataset on text-and-pose guided image generation. † taken from [142].

We use the DeepFashion Multimodal dataset proposed by Text2Human [51] in which a segmentation map and text description accompany each image. We train on the resolution 512 × 352. We follow Text2Human's data split and crop the generated images into 512 × 256. The baseline methods [51, 142] cannot control clothing colour, which would hugely affect evaluation scores. For a fair comparison, we train our models without clothing style image embedding.



(a) “The gentleman is wearing a long-sleeve shirt with floral patterns and short pants with pure colour patterns.”

(b) “The woman wears a short-sleeve shirt and short skirt in pure colour.”

(c) “The lady is wearing a sleeveless shirt, a short pant, and a hat.”

Figure 4.6: *(Zoom in to view full* 512 × 256*)* resolution. (a) We generate a variety of clothing types and texture patterns directly from SMPL pose parameters while Text2Human has additional stage to create parsing map from pose (DensePose[34]). (b) Text2Human tend to generate blended crossed legs when the parsing map overlapped. (c) Using vocabulary outside of Text2Human limited dictionary can result in defective parsing map and hence erroneous final image.

Table 4.3 shows our method achieving the best FID score against the baselines. Next, we perform some qualitative analysis. HumanDiffusion[142] does not provide code to reproduce their results, but their paper shows blurry images with colour saturation. Both us and Text2Human can generate high quality images, as shown in Figure 4.6a and Figure 4.9, but there are a few shortcomings with the latter. Text2Human cannot generate images directly from the pose, and it must first generate a parsing map from the pose and text. As also observed by [142], we found that they systematically exhibit blended crossed leg when parsing map overlapped (Figure 4.6b). Parsing maps can indeed introduce gender bias, particularly when certain features, such as long hair, are associated with female identities. Also, Text2Human has limited text capability. Their model was trained on categorical labels and added text-to-category mapping later. Therefore, vocabulary falling outside of their dictionary can generate the wrong parsing map. This is demonstrated in Figure 4.6c. The word *pant* rather than *pants* was used in the text prompt, and that causes the skin (green) and top clothing (white) to smear into bottom clothing (gray). The following sections will show our superior visual and text prompting capability.

4.4.2 Pose Transfer

We use DeepFashion[156] In-shop Clothes Retrieval dataset for the pose transfer task. Using the given train-test split of individual images (48675 and 4039, respectively), PATN[155] proposes a pose transfer dataset of about 102k image pairs for training and 8570 pairs for testing. Given our model architecture's flexibility to support individual and image pairs in training, we combine both as our training dataset. As the Inshop subset does not provide a text description of images, we use the text labels from the Multimodal subset, which cover most of the samples in Inshop. We resize the image to 256×176, maintaining the same aspect ratio. We also combined the fine-grained segmentation map from both subsets. However, a small number, about 5% of Inshop test image pairs, either have incomplete text or segmentation maps or do not contain humans; we excluded these from the test set. We evaluate our and reference methods using the same reduced test set to obtain the fair quantitative results in Table 4.4.

Method	**FID↓**	**LPIPS↓**	**SSIM↑**
ADGAN[74]	20.025	0.2289	0.6856
PISE[140]	17.799	0.2273	0.6781
DPTN[144]	16.686	0.2192	0.6958
CASD[150]	10.439	**0.1777**	**0.7131**
UPGPT(ours)	9.427	0.1886	0.6970
NTED[91]	**8.813**	0.1814	0.7011
†UPGPT(ours)	7.876	0.1766	0.7276

Table 4.4: Quantitative results on pose transfer task. † compare the generated images against images reconstructed by VAE.

Although our method is not designed explicitly for the pose transfer task alone, we near state-of-the-art results; we found that small faces in our generated images can appear blurry due to the inadequacy of VAE in capturing rich details in small faces. In other words, an image x reconstructed $\mathcal{D}_{\mathcal{I}}(\mathcal{E}_{\mathcal{I}}(x))$ by VAE can have a blurry face even if our model produces a perfect image latent. To confirm this, we compare our generated images against the images reconstructed from the ground truth images rather than the ground truth images, and the scores improve significantly to top the performance table.

Apart from that, our method produces realistically looking images and excels in utilising all modalities when information in the source image is incomplete or incorrect. This is best demonstrated in the pose transfer task in Figure 4.7, where the jacket in the source image (1) is removed from the target image (9). Even assuming the person still has their jacket on, existing methods (2-6) often fail to distinguish between the jacket and the top wear, blending the style and texture to create incorrect clothing. In contrast, our results (7) show clear distinguishment, resembling the source image appearance. UPGPT blocks out the jacket in the image generation process by conditioning on the context text of the target image, creating our final pose transfer result (8) that resembles the ground truth target image (9).

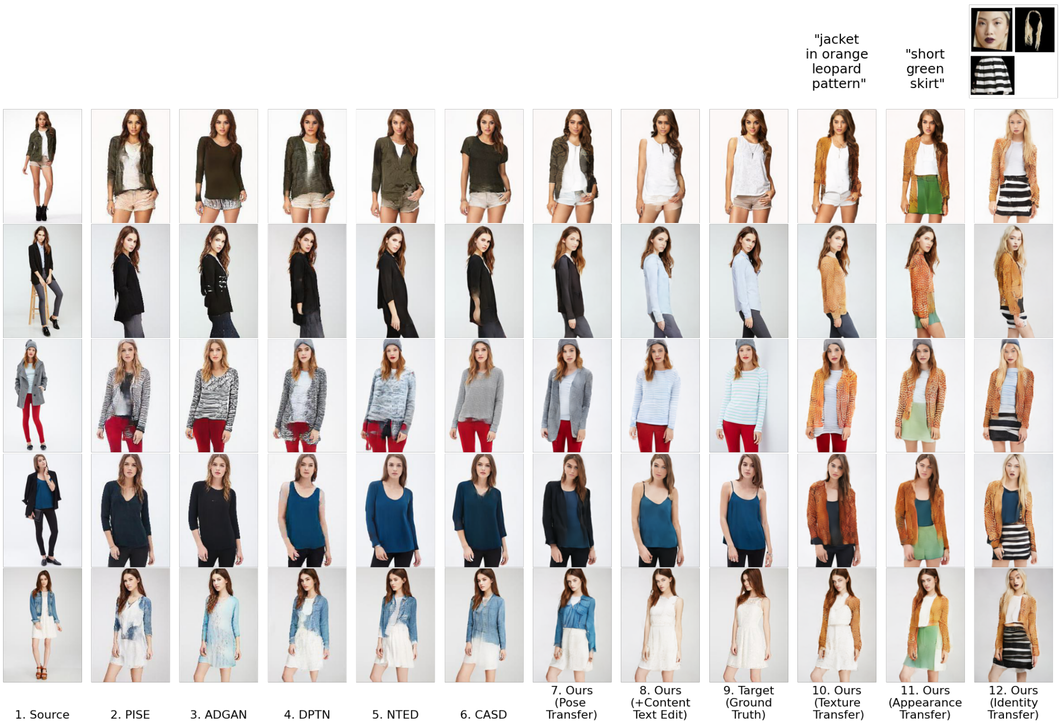
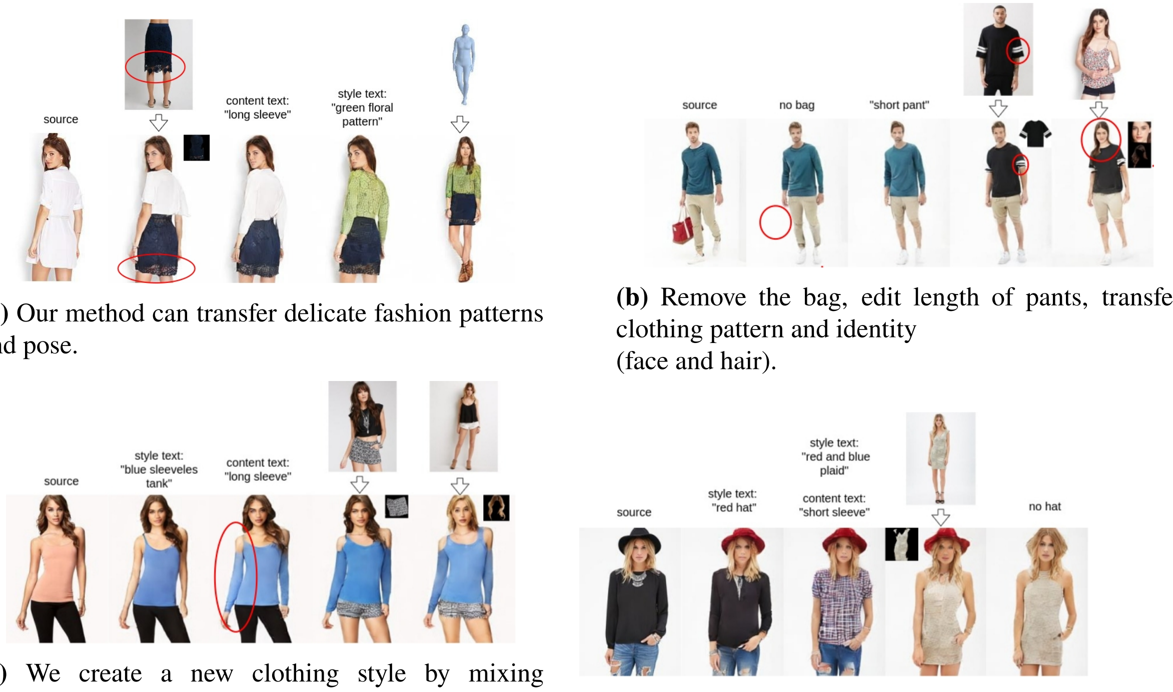

"jacket in orange leopard pattern"
"short green skirt"
1. Source
2. PISE
3. ADGAN
4. DPTN
5. NTED
6. CASD
7. Ours (Pose Transfer)
8. Ours (+Content Text Edit)
9. Target (Ground Truth)
10. Ours (Texture Transfer)
11. Ours (Appearance Transfer)
12. Ours (Identity Transfer)

Figure 4.7: *(Zoom in to view)* Pose transfer from (1) source image into the (9) pose target in which the jacket is removed. Reference methods PISE[140], ADGAN[74], DPTN[144], NTED[91], CASD[150] blend the top wear and jacket to generate the wrong clothing (2-6), while ours (7) create clear separated jacket from top wear, matching the source image appearance. Conditioning on the content text that correctly describes the target image, we create the final pose transfer result in (8) matching the ground truth (9) appearances. (10) and (11) show we can perform consecutive texture and appearance transfers with texts. In (12), we show how to perform texture and identity transfer using style images while still conditioning on the previous style text edit.

4.4.3 Flexible Image Editing

Columns (10-12) in Figure 4.7 demonstrate the flexibility of our fine-grained control method. From (7), we replace the jacket style image with the style text “jacket in orange leopard pattern” to perform texture transfer (10). Our style text has good zero-shot capability, and we can use words like zebra, pandas, and oceanic instead of colour. Then, we change the context and style texts in (11) to replace bottom wear with a short green skirt, changing the texture and clothing type *i.e.* appearance edit. Please note that the jacket from (10) remains in (11), showing that our approach allows for consecutive editing. This is a significant improvement from existing methods [7, 91, 150] that have demonstrated only to transfer appearance from a single image reference. In contrast, we can mix different modalities from different sources to perform flexible and fine-grained control across clothing type, texture, or both. Although it is convenient to use text to change clothing types and colours, some things are difficult to describe in words *e.g.* specific clothing patterns or the face of a particular person. Therefore, our methods also support using styles images for editing and identity transfer, as shown in (12). The pipeline of going from (1) to (12) demonstrates we can mix and match different modalities - pose, style, content text, and style images to achieve excellent fine mix-and-match in generating and editing person images.

We provide more image editing examples using a model trained on the low resolution 256×176 images. Our method allows for simultaneous and consecutive image editing using multimodality from multiple sources. Some baseline pose transfer models can perform only a single appearance transfer (clothing texture, shape, or face) from a single reference image, but we could do much more. In Figure 4.8a, we demonstrate the capability of our method to transfer a delicate clothing style, followed by text edits and pose transfer. We can also remove objects (bag in Figure 4.8b, hat in Figure 4.8d). Figure 4.8c shows how we can create a new clothing type not from the dataset by mixing "sleeveless tank" in the style text and "long sleeve" in the content text. Baseline methods are limited to fashion transfer of the same type *i.e.*, top wear to top wear. Still, we can do any combination of fashion transfer, such as replacing a shirt and pants with a dress, as shown in Figure 4.8d.


source
content text: "long sleeve"
style text: "green floral pattern"
source
no bag
"short pant"
source
style text: "blue sleeveles tank"
content text: "long sleeve"
source
style text: "red hat"
style text: "red and blue plaid"
content text: "short sleeve"
no hat

(a) Our method can transfer delicate fashion patterns and pose.

(b) Remove the bag, edit length of pants, transfer clothing pattern and identity (face and hair).

(c) We create a new clothing style by mixing “sleeveless tank” in style text with “long sleeve” in context text. We can also provide fine-grained transfer of only the hair.

(d) Replacing two garment pieces (shirt and pants) with a single dress.

Figure 4.8: Starting from the source image in the left, we perform step-by-step consecutive image editing from multiple multimodal sources.

4.4.4 Comparing Text2Human

Overall, Text2Human and our method, UPGPT, can generate high quality images; we display examples of both results and ground truth in Figure 4.9. Some of Text2Human's images may appear smaller because of the padding they added to the dataset, while we use unmodified DeepFashion Multimodal images. However, Text2Human has two major limitations that can affect the overall visual perception - (1) systematic error in crossed legs and (2) poor gender and pose disentanglement.

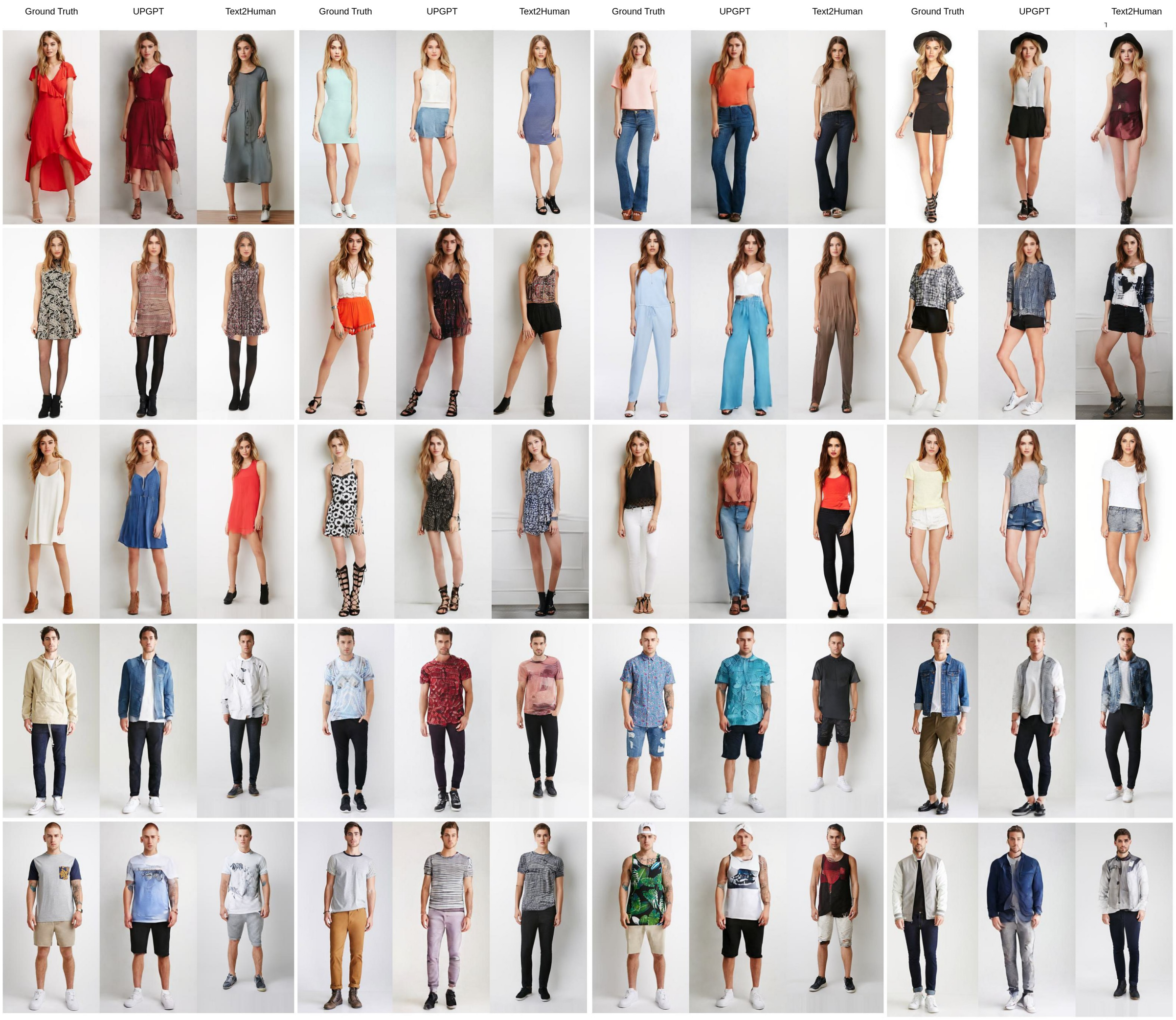


Figure 4.9: (Zoom in to view full resolution) Both UPGPT(our method) and Text2Human can generate high quality images.

4.4.5 Blended Crossed Legs

Figure 4.10 shows systematic error in the legs when crossed and blended in the parsing map and results in the same in the generated images. We avoid this problem by using the SMPL model as pose guidance which contains 3D body pose information.

4.4.6 Poor Gender and Pose disentanglement

In Text2Human, the body appearance ties closely to the parsing map. Figure 4.11a shows that Text2Human generates two parsing maps - male and female from the pose. There is very little difference between them apart from the hair length. Due to incompatible body proportion, Text2Human females' overall appearance (Figure 4.11a) have subtle unnaturalness compared to ours in 4.11b. Although we use only the female SMPL model to train our model, our model can generalize the genders well yet provide good disentanglement between gender and pose. The gender bias in Text2Human can be further shown in Figure 4.12 where short haired parsing maps often result in a male face, which doesn't occur with our

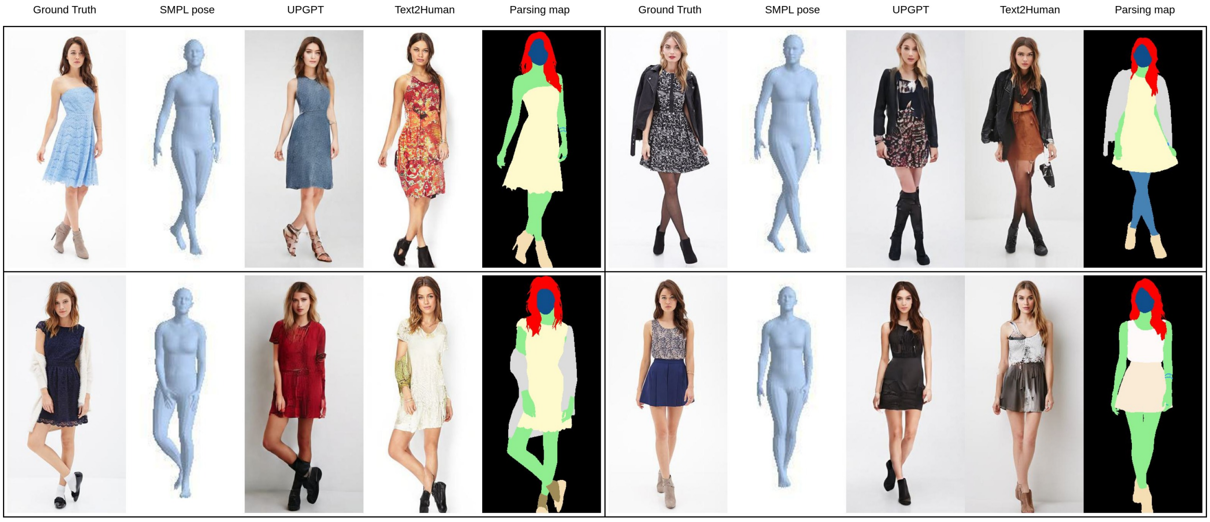
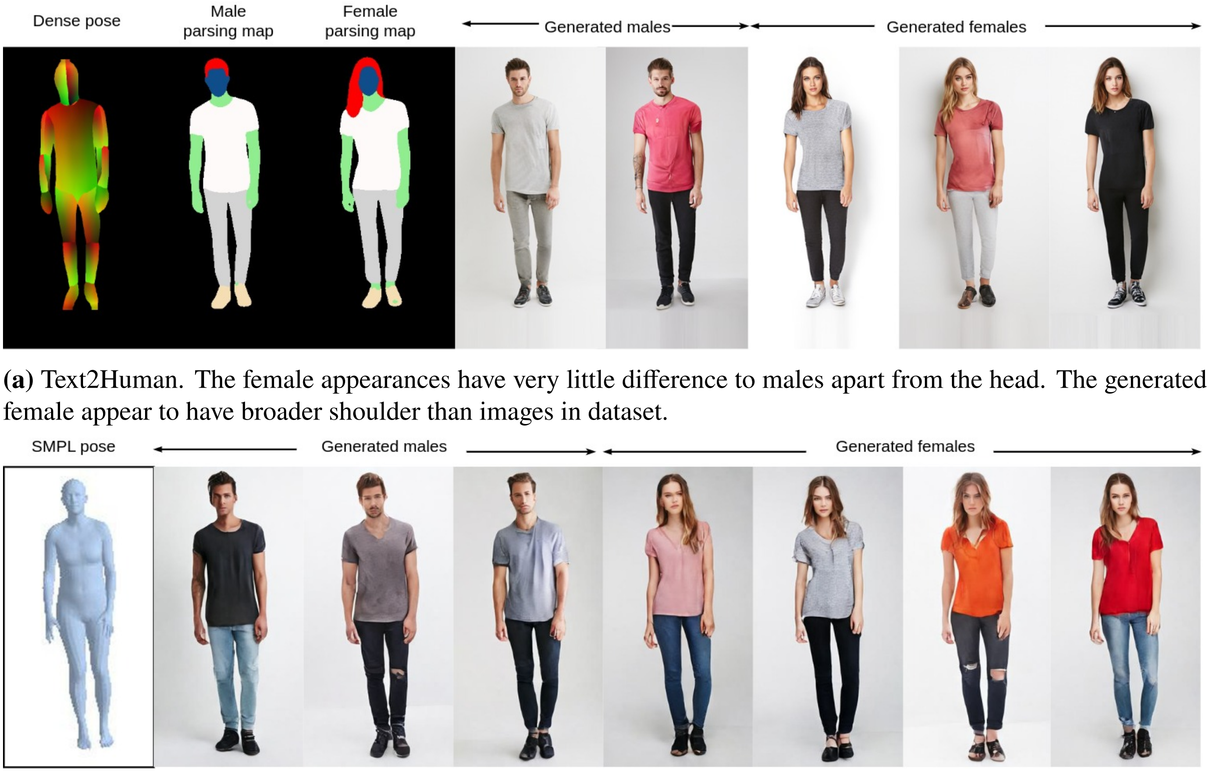


Figure 4.10: *(Zoom in to view)* Text2Human often generates erroneous crossed legs from parsing map. Our method avoids this problem by using the SMPL model as pose conditioning.

approach.



(a) Text2Human. The female appearances have very little difference to males apart from the head. The generated female appear to have broader shoulder than images in dataset.

(b) UPGPT. People generated from the same pose look more natural for their genders.

Figure 4.11: Our method provides better disentanglement between pose and gender.

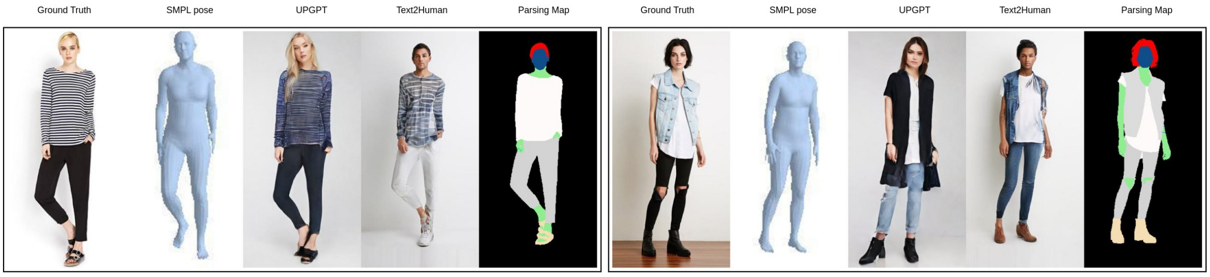


Figure 4.12: Text2Human tends to generate male faces from parsing maps with short hair.

4.4.7 Pose and Camera View Interpolation

We demonstrate our approach's superior pose capability and disentanglement with simultaneous pose and camera view interpolation as shown in Figure 4.13. The pose interpolation can be performed by linear interpolating SMPL parameters between two poses. To our best knowledge, this is the first demonstration of pose interpolation within the human image generation literature. This paves way for parametric camera pose control and interpolation in video generation in *Chapter 6: Guiding Camera Motion in Video Diffusion Transformer*.

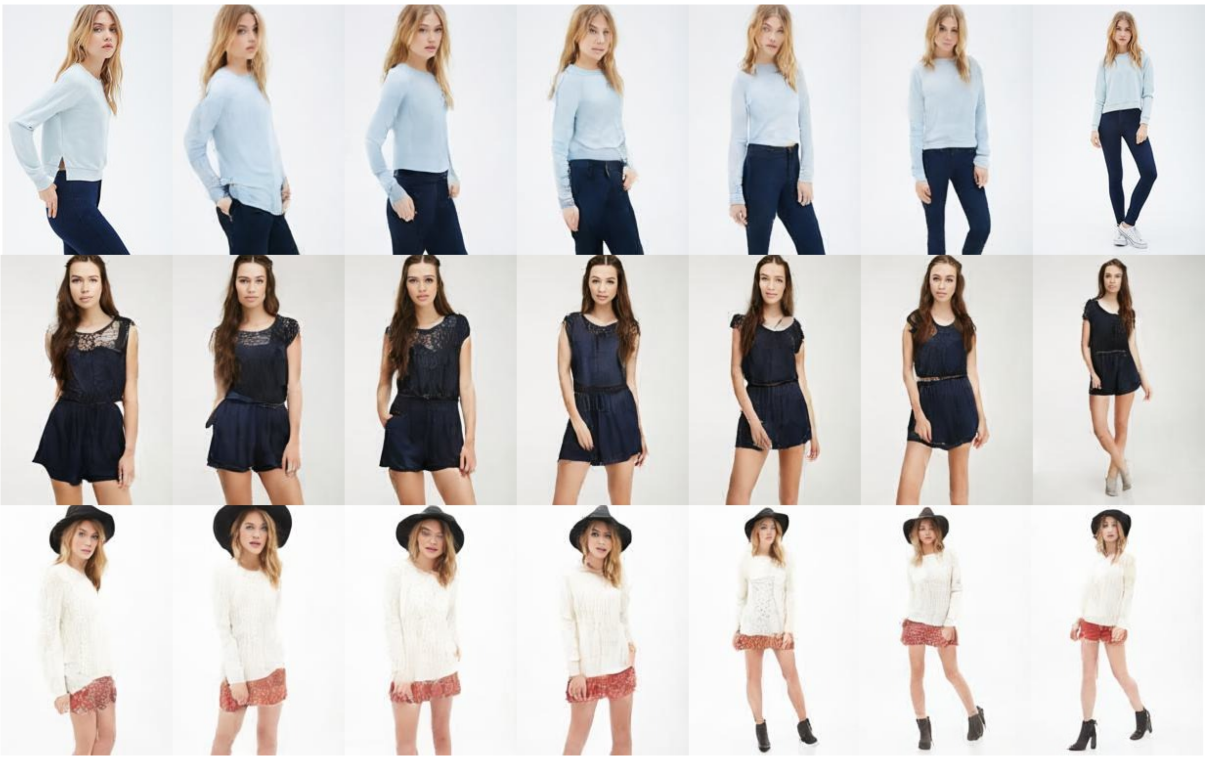


Figure 4.13: Complex hand and camera movement achieved using linear pose interpolation. By using poses from images at the left and right ends, we interpolated the poses to create images in between.

4.4.8 Ablation Study

We performed experiments to explore the importance of the reinforced pose masks (RPM) and evaluated their performances as shown in Table 4.5. Qualitatively, without any form of RPM, the person in generated images looks visually similar to other masks apart from the occasional horizontal offset. We explore two methods: a bounding box and a mask derived from the SMPL render. Including a bounding box as a mask improves the scores compared to not having one. A further approach is to use the silhouette mask created from SMPL rendering as the segmentation input. However, the derived mask is less accurate, and the result is slightly worse than using a silhouette mask estimated from 2D images.

Reinforced Pose Mask (RPM)	**FID↓**	**LPIPS↓**	**SSIM↑**
w/o RPM	10.176	0.2670	0.6146
w bounding box	10.100	0.2447	0.6254
w SMPL rendering	**9.245**	0.2149	0.6604
w silhouette mask	9.427	**0.1886**	**0.6970**

Table 4.5: Quantitative results of ablation on reinforced pose mask.

4.5 Limitations

Apart from the blurry small faces discussed in Section 4.4.2, one of our method's limitations is that clothing textures sometimes do not match the style image *e.g.*, the stripes can have different thicknesses. This is due to the limitation of the CLIP image encoder, which does not necessarily capture fine-grained spatial detail but focuses on the overall colour response.

4.6 Conclusion

In this work, we proposed UPGPT, the first universal method to perform unified person image generation, editing, and pose transfer tasks. Unlike existing methods that require masks for editing, our maskless approach provides a convenient way of fine-grained person image editing using a combination of modalities. We achieved competitive pose transfer results in comparison to the state-of-the-art methods. Also, we overcame the inadequacy of SMPL pose estimation to incorporate it into our model to improve pose disentanglement and demonstrate the first simultaneous pose and camera view interpolation in pose-guided image generation literature.

While our approach of integrating pose and image tokens enhances generative model capabilities, introducing these tokens alongside pre-existing text tokens, requires retraining to achieve proper alignment between these diverse embeddings. This retraining process is computationally expensive, as it involves significant model adjustments to ensure effective multimodal integration. In the next chapter, we will address this challenge by employing adapters, which enable us to incorporate new modalities and controls without full retraining.

Chapter 5

Avoiding Mode Conflict with Unified Pose-Visual Diffusion Adapter

In this chapter, we delve into the use of adapters — lightweight, modular models, to incorporate multiple conditioning inputs — text, visual, and structural—in a balanced manner while avoiding mode conflict. We achieve this by isolating the structural and visual conditioning within the adapter and harmonise its control signal connecting the base image diffusion models. This modular approach allows for partial training only on the adapter, and avoid full training of the base model, thus significantly reducing computational load.

Supplementary images and project demo are available from project page

`https://soon-yau.github.io/visconet/`

5.1 Introduction

Adding new control to T2I diffusion models can be achieved by concatenating tokens from additional modalities, like pose tokens, with text tokens, providing cross-attention conditioning, as shown in the last chapter. However, this can lead to challenges due to the incompatibility between new modality embeddings and text tokens, and it increases cross-attention input size, which requires extensive retraining [17, 53, 67, 142], demanding substantial datasets and computational resources. Recognising this challenge, recent methodologies like T2I-Adapter[77] and ControlNet[143] have introduced a pragmatic approach. They integrate a lightweight adapter branch[44] to encode structural conditioning information, such as pose or segmentation maps, onto a frozen pre-trained T2I LDM backbone. In a more recent development, methods such as IP-Adapter[136] and MasaCtrl [11] extend this concept by introducing visual conditioning capabilities. However, they cannot control pose independently and require a separate structural adapter, introducing additional computational complexity to the overall architecture. However, training adapters on smaller and disparate datasets may introduce a domain gap with the frozen LDM model. As noted by [11, 53], this conflict between branches can manifest in the model's inability to generate people following specified text and pose conditions. The situation may be exacerbated when multiple adapter branches are employed. Our work addresses this issue by striving to develop a lightweight adapter that accommodates both pose and visual conditioning. This singular adapter aims to excel in a spectrum of human image generation tasks, unifying functionalities currently achievable through utilising distinct models.

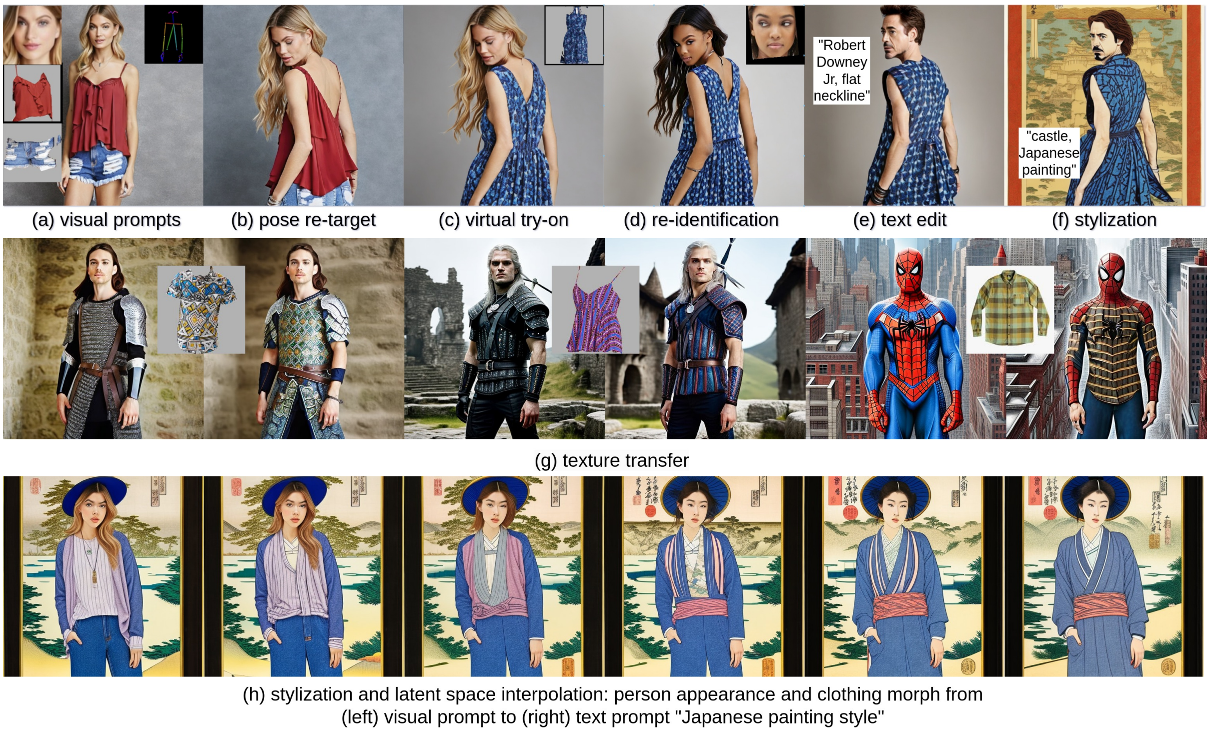

"Robert Downey Jr, flat neckline"
"castle, Japanese painting"
(a) visual prompts
(b) pose re-target
(c) virtual try-on
(d) re-identification
(e) text edit
(f) stylization
(g) texture transfer
(h) stylization and latent space interpolation: person appearance and clothing morph from (left) visual prompt to (right) text prompt "Japanese painting style"

Figure 5.1: Our proposed **Visconet** demonstrates broad versatility in multimodal human image tasks including visual prompts, pose re-target, virtual try-on, re-identification using either text or visual prompt, text prompt, texture transfer, stylisation and latent space interpolation to perform human morphing.

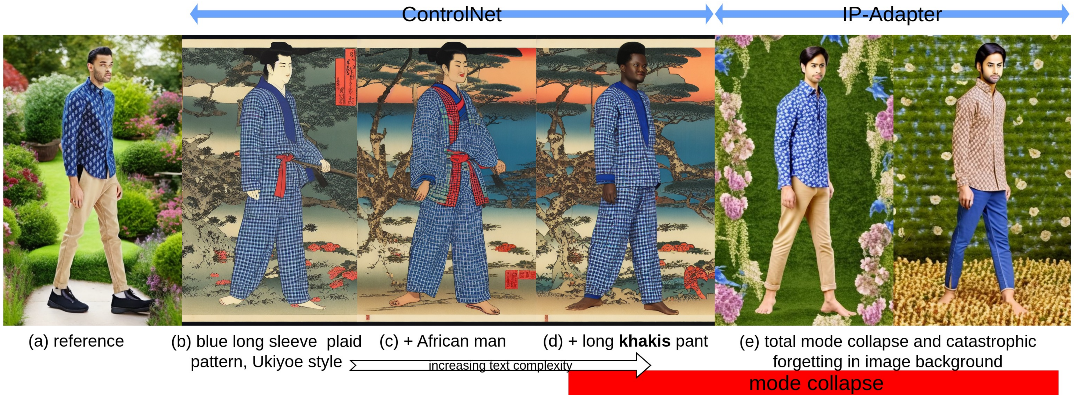

ControlNet
IP-Adapter
(a) reference
(b) blue long sleeve plaid pattern, Ukiyoe style
(c) + African man
(d) + long **khakis** pant
(e) total mode collapse and catastrophic forgetting in image background
increasing text complexity
mode collapse

Figure 5.2: To motivate our work, this figure illustrates how increasing text complexity in ControlNet [143] can expose (c) domain gap and eventually lead to mode conflict in (d). IP-Adapter [136] also exhibits (e) catastrophic forgetting, resulting in the inability to generate a rich background. Both show the concept of bleeding by assigning the wrong colour to clothing garments.

We illustrate the domain gap and conflict of adapters in Figure 5.2 where ControlNet attempts to reconstruct reference images with increasing text complexity from (b) to (d). Figure 5.2c shows a sign of domain gap as dark-skinned people were not typical in ancient Japanese drawing (Ukiyoe style). We continue adding “khaki”, a more modern term, into the text prompt. The complexity eventually exposes the domain gap between ControlNet and T2I. As a result, ControlNet resorts to generating realistic people that it learned from its small training data (Figure 5.2d). This is a phenomenon we call **mode conflict (MC)**. Mode conflict has existed since GANs [31] but has not been discussed recently despite widely affecting recent diffusion model-based adapters. We are the first to study mode conflict in an adapter-based diffusion model systematically. There is currently no effective mechanism to control and

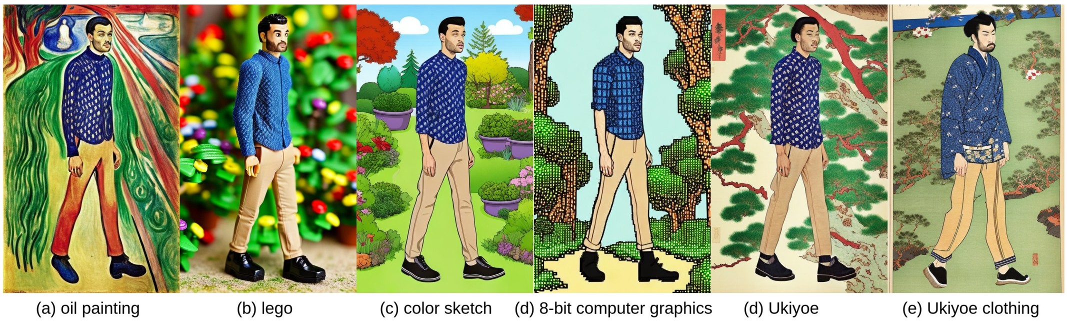


Figure 5.3: Our method retains generative power of the T2I backbone in (a)-(d) various image styles and rich backgrounds while maintaining the person and clothing appearance, assigning correct clothing colours. In (e), we can control the level of stylisation to expand it to the clothing styles.

manage this conflict; only when one of the conflicting texts, i.e., khaki or Ukiyoe style, is removed will it escape the stuck mode. Unfortunately, this restricts the image content that can be generated. The general solution is to train a more extensive dataset to close the domain gap. HumanSD[53] compiled a 1M image dataset, up from ControlNet's 200k, while IP-Adapter uses 10M[136] and more recent Hyperhuman [67] ballooned to 340M! This is an inefficient use of computing resources, and as we will show, this is insufficient to eradicate mode conflict completely. Conversely, training on a limited dataset may lead to overfitting and, consequently, catastrophic forgetting. This is evident in the model's diminished ability to generate diverse individuals, varied image backgrounds, or encompassing artistic styles as depicted by the input prompts. In contrast, our method trains only on about 50K images, many orders of magnitudes smaller than reference methods.

In this work, we propose a novel architecture extending ControlNet [143], which we call **ViscoNet** (**Vis**ual **Co**ntrol**N**et), bridging and harmonising visual and text conditioning. Our method's ability to fuse and control the balance of both text and visual conditioning unlocks unparalleled versatility in HIG, which includes pose re-target (transfer), virtual try-on, person re-identification (face swap) with both text and visual, image stylisation, textile transfer, and visual-text latent space interpolation to achieve morphing as shown in Figure 5.1. The summary of our contributions:

1. A lightweight one-branch adapter architecture for structural and visual conditioning.
2. Excellent ability to control and harmonize text and visual prompts, significantly mitigating mode conflict and empowering various HIG capabilities.
3. Our training with feature masking effectively preserves the backbone model's generative capabilities on a small dataset, mitigating catastrophic forgetting.

5.2 Related Works

Visual Conditioning. Image personalisation methods [28, 98] explore finetuning text vocabularies to define specific identities. [13, 29] follow the same idea, while [14, 49, 104] leverage large-scale upstream training to eliminate the need for test-time finetuning. These methods use text to control visual aspects rather than images as input conditioning. In HIG, UPGPT [17] pioneered visual conditioning in the

T2I diffusion model by concatenating visual tokens alongside text tokens and pose tokens. However, it changes the model architecture and unable to re-use the pre-trained model weights.

Adapter. More recently, adapter modules and lightweight models have been added to pre-trained, frozen diffusion models for faster finetuning requiring less data; among them are ControlNet [143], T2I-Adapter[77]. However, as they add the learned feature spatially to the UNet's multi-resolution layers in the diffusion model, the control signals are limited to the spatial dimension. Although the T2I-Adapter demonstrates the use of reference images for visual conditioning, it is constrained to the overall artistic style of the image. MasaCtrl[11] is a tuning-free method that injects masked self-attention features from a reference image in the T2I denoising step. IP-Adapter[136] uses a separate cross-attention map for image conditioning to be added to the textual attention map. The balance can be adjusted using a weighted average between the two attention maps. Both IP-Adapter and MasaCtrl are conditioned on a single image for a global image, lacking fine-grained visual conditioning. Uni-ControlNet[147] supports both global and local image but still employs dual-branch design. InstantID[123] is based on IP-Adapter's architecture, with the main difference being swapping the CLIP image encoder with a specialized face encoder. While they focus on human face, our method exhibits a broader capacity, generating full human body with higher complexity.

Dancing Avatar. This group of models re-purposes T2I into image-only-conditioning to reconstruct humans for dancing avatar videos faithfully. They sacrifice the T2I's text capability and are not directly comparable to our method. Nevertheless, we scrutinize their pose-and-visual methods. Disco [124] uses ControlNet to inject static image background signal. To ensure visual consistency of the moving foreground person, it applies a visual signal to cross-attention of UNet in image-to-image SD variant [83], which requires re-training. MagicAnimate [132] and AnimateAnyone [45] use a dedicated adapter branch to encode visual information to be fused with UNet using cross-attention.

Overall, existing methods [11, 45, 77, 124, 132, 136, 143, 147] employs multiple adapters for simultaneous pose (e.g. ControlNet) and visual control (e.g. IP-Adapter). Our method introduces improvements over a single ControlNet to offer both pose and visual control, saving computational requirements and potentially mitigating conflicts introduced by multiple branches.

5.3 Method

5.3.1 Preliminaries

Stable Diffusion (SD), a backbone LDM [95], and a ControlNet model [143] are shown in the left and right block in Figure 5.4. SD uses a UNet[96] as the denoising network and progressively refines the input noise into latent variables that can be reconstructed into realistic synthetic images, relying on understanding intricate image distributions. The words within a text prompt are decomposed into smaller subword units and tokenised and encoded with a CLIP [86] text transformer [120]. The text embedding is injected into the cross-attention layers in UNet, serving as the sole conditioning in image generation. The loss function of the LDM is:

$$\mathcal{L}_{\mathcal{MSE}} := \mathbb{E}_{z,c,t,\epsilon\sim\mathcal{N}(0,1)}\left[\|\epsilon - \epsilon_\theta(z_t, t, c)\|_2^2\right] \tag{5.1}$$

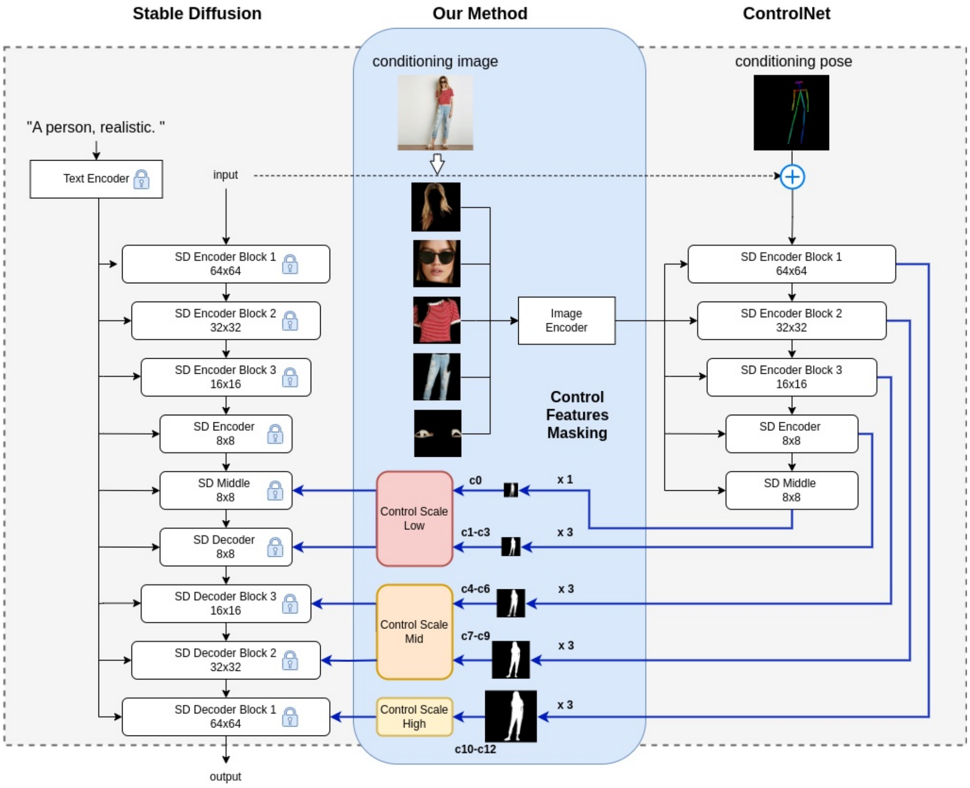


Figure 5.4: Architectural diagram showing our contribution concerning backbone LDM and ControlNet layers. We omit time embedding, zero convolution, and some blocks from the ControlNet diagram [143] for simplicity.

where c is the text conditioning token, t is the diffusion time step, and z is the latent variable (denoted as input in Figure 5.4).

Instead of training from scratch, ControlNet [143] adds a learnable branch parallel with a now frozen pre-trained LDM, as shown on the block on the right in Figure 5.4. The branch consists of an identical LDM UNet encoder copy, sharing the same latent noise input and text embedding. It learns to control pose conditions by adding skeleton image features into the latent noise input at the branch input. ControlNet generates structural control signals and adds them to the SD decoder across multiple spatial resolutions.

5.3.2 Replace Text with Visual Prompt

That ControlNet's sharing of the exact text embedding with the LDM is unnecessary when learning the structural condition. Their mandatory use of text-image pairs in training places an excessive burden on data collection and annotation to the specific image and text styles. The text entanglement also increases potential conflict between the branch and LDM [53]. Therefore, in our architecture, we remove the text prompt from ControlNet to sever the entanglement and replace it with a visual prompt. Unlike [124, 136] that use a single reference image for overall visual conditioning, we use multiple images consisting of segmented body parts (e.g. hair, face, top clothing, bottom clothing) for fine-grained visual control on individual clothing garment pieces (Figure 5.1a-e).

The de-facto image encoding method for the diffusion model uses a CLIP image encoder to extract a global image embedding, but this is insufficient in capturing intricate image details. Therefore, we utilise the larger dimension local CLIP embedding before the pooling layer. We project the local CLIP embedding of individual images using a linear layer into length N and concatenate them like the text tokens they replace. N can be adjusted based on the number of conditioning images and the text token length limit of ControlNet, and we use N=8 in this work. The linear layer consists of only 2K parameters. It is the only additional trainable parameter introduced by our method, 10,000 times fewer than IP-Adapter's 22M parameters. Controlling all the human information (pose and visual appearance) within the single adapter branch creates effective disentanglement from the LDM to avoid conflict. For example, "ripped jeans" may conflict with "Picasso painting"; having both in a text prompt could trigger mode conflict in ControlNet. Instead, we can avoid this by removing "ripped jeans" from the text prompt and replacing it with an image in the visual prompts.

5.3.3 Control Feature Masking

DeepFashion[156] is a popular and de-facto dataset in many HIG tasks in machine vision literature. However, all the images consist of plain studio backgrounds, which will overfit and severely restrain LDM generative capability, resulting in dull and bland image backgrounds. This phenomenon is observed with IP-Adapter [136] in Figure 5.2e despite it being trained on a large dataset of 10M images. To tackle this issue, we apply a binary human silhouette mask to the control signals originating from our adapter branch before injecting it into the LDM. This eliminates unwanted image backgrounds leaking into and causing overfitting in the LDM. The disentanglement between foreground people and background contributes to reducing the image domain gap with the LDM. This improvement empowers our method to harness LDM text capability to generate vibrant backgrounds in various artistic image styles despite training only on a relatively tiny dataset (only 52K) of images with plain backgrounds. In Section 5.5, we delve into further analysis, demonstrating that feature masking is essential during training rather than used solely during image sampling.

The feature mask is also applied to the LDM loss function (Equation 5.1) at the *output* in Figure 5.4. The training loss backpropagates via the frozen LDM to train the model. This approach is akin to [17, 53], although they use it to assign weight loss to different body segmentation parts rather than masking a region entirely. We add masking to the LDM loss function 5.1:

$$\mathcal{L}_{\mathcal{MSE}} := \mathbb{E}_{z,c,v,t,\epsilon\sim\mathcal{N}(0,1)} \left[||\mathcal{M} \odot (\epsilon - \epsilon_{\beta}(z_t, t, c, v)||_2^2) \right] \tag{5.2}$$

where ϵ_{β} is the model, v is the image embedding, $\odot$ is the element-wise multiplication, and $\mathcal{M} \in \mathbb{R}^{H,W}$ is the binary mask resized to resolution (H, W) of the LDM output. Although text conditions are not used in the model, they are used by LDM in training and, thus, are included in the equation.

5.3.4 Harmonising Text and Visual Influence

The versatility of our approach in performing diverse human image generation tasks arises from its ability to seamlessly integrate and regulate the balance between visual and textual conditioning. We achieve this by multiplying scalar values $[0.0, 1.0]$ with the control features. Scaling control signals is commonplace in adapter-based approaches, but our novel model architecture unlocks unprecedented effects not observed in existing methods. Despite innovations adopted to reduce data conflict between the adapter and the LDM, mode conflict can still happen in challenging image styles. In this scenario, we can decrease the control signal strength to weaken the visual prompt strength to escape the mode conflict. As we will show, the application of this approach has no discernible impact on mitigating mode conflict in ControlNet [143] and other structural conditioned models [53, 77]. This is attributed to the fact that its control signals exclusively influence pose conditioning, whereas the root causes of the conflict lie in the image domain gap and text entanglement. In contrast, our innovative architecture, which involves the separation and subsequent bridging of text and visual conditioning, empowers us to dynamically adjust their balance, thereby enabling latent space interpolation (Figure 5.1h) and eliminating mode conflict.

On the other hand, IP-Adapter [136] supports visual prompts, and it can adjust the text-visual balance by changing the scales of the respective cross-attention map, but the effect is global to the image. For example, tipping the balance away from the visual prompt of a realistic photo of a man towards the text prompt “a girl, Chinese ink painting” would result in the global transformation of a modern man towards a Chinese girl wearing period Chinese clothing in Chinese ink painting style. Our method can apply different scaling at each multi-spatial resolution to customize at different image levels. This is demonstrated in Figure 5.1f, which depicts only the image’s artistic style while retaining the person’s identity and appearance, and Figure 5.1h, which shows the morphing only of the person, leaving the background essentially unchanged.

For the sake of discussion, the 13 individual control strength scales (c0-c12) can be roughly grouped into three blocks - Low Blocks (LB), Mid Blocks (MB), and High Blocks (HB) arranged hierarchically from low to high spatial resolution. We can adjust their values separately to create different effects. Through experimentation, we observed that LB exerts negligible influence and can effectively be set to 0. The MB is the most influential in overall visual appearance styles among them. HB regulates fine image texture, aligning with our expectations for image hierarchy control. Setting HB alone yields the notable outcome of transferring only the texture of the visual prompt (Figure 5.1g). We can also constrain our control to pose only by setting c4 to 0.5 while leaving others 0.0, allowing using text prompts to control the whole person’s appearance.

5.3.5 Training Setup

To train the model, we employ 52K-images DeepFashion In-shop Clothes Retrieval dataset [156] and adopt the train-test split proposed by [155] for the pose transfer task, padding the images to the size of 512×512. Pose information is extracted using OpenPose [12] to create body-and-hand skeleton images, and we use pre-segmented fashion images from [17]. We employ a simple text prompt of “a person” for all the images. This serves two purposes: first, the neutral description avoids potential conflicts with the LDM, proving our method does not need to annotate text to match the style of the LDM carefully. Secondly, it acts as an unconditional text embedding, enabling users to amplify the desired visual effect using positive prompts, negative prompts, and guidance scales[22].

Many adapter models are based on pre-trained SD or similar-sized models. Thus, we also performed our experiments using SD2.1[112] for a fair comparison. We initialize our adapter branch by copying frozen weights from the SD. However, since the cross-attention input has shifted from global CLIP text embedding to local CLIP image embedding, we re-initialize the weights in the cross-attention layer at the start of training. All weights in the SD, CLIP text, and image encoders are frozen. We use CLIP image encoder *clip-vit-large-patch14*[47]. We trained the model on a single desktop GPU GTX 3090 for 2 epochs, using a batch size of 4 with four gradient accumulations per batch, resulting in an effective batch size of 16. We retained the remaining configurations from [143].

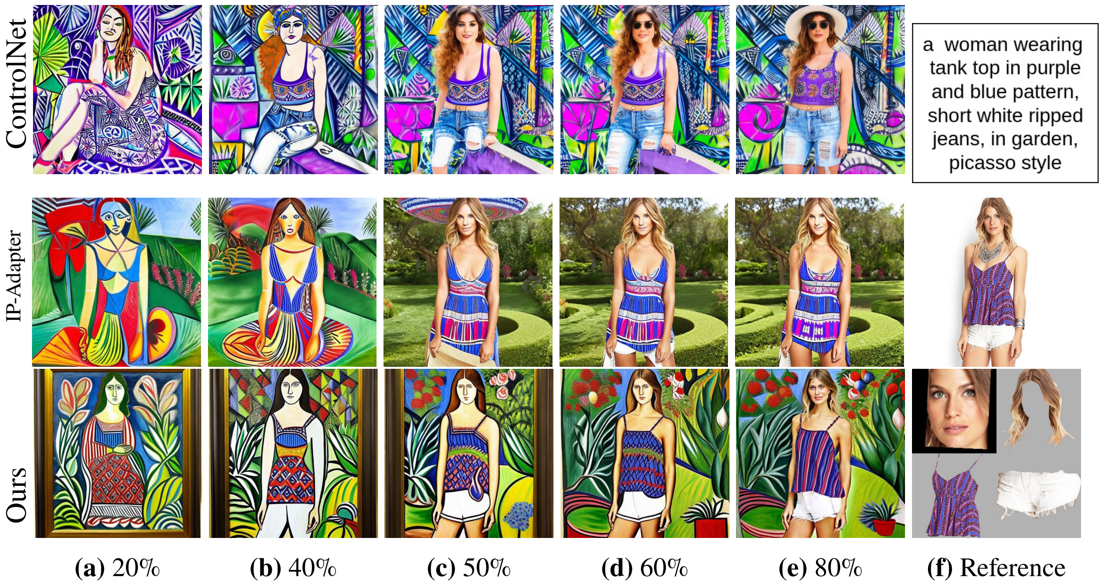
5.3.6 Image Resolution

We use an image resolution of 512 × 512 for all the experiments in this chapter. In our experiment, we utilised $^3/_4$ length to full-body images, resulting in smaller human faces within the images. The stringent demand for high pixel density per latent variable can lead to suboptimal face construction[17] compared to high resolution face images generated by [123, 136]. This inherent limitation is a characteristic drawback of the LDM rather than a weakness of our method.

5.4 Experiments

In Section 5.4.1, we perform an in-depth study of the effect of control strength on mode conflict as observed in image artistic styles. We show that visual prompt methods (IP-Adapter and ours) effectively reduce mode conflict compared to ControlNet. Then, in Section 5.4.2, we perform further, more challenging experiments in person re-identification to show our method has superior text-visual harmonisation capability compared to IP-Adapter. Lastly, we performed large-scale human evaluation in Section 5.4.4 to prove our image quality over SOTA HIG models.

5.4.1 Mode Conflict and Control Strength


ControlNet
IP-Adapter
Ours
a woman wearing tank top in purple and blue pattern, short white ripped jeans, in garden, picasso style

(a) 20% **(b)** 40% **(c)** 50% **(d)** 60% **(e)** 80% **(f)** Reference

Figure 5.5: Effect of control strength (%). Compared to ControlNet and IP Adapter, our method can escape mode conflict faster, generating a harmonious image style while maintaining good visual control.

In this section, we examine the prevalence of mode conflict and its impact on existing structural and visual adapter models compared to our proposed model. In Figure 5.5, conditioned on the same human pose, we generate images of *Picasso style* at different control strengths. ControlNet does not have visual input, thus we use text to describe the person's appearance and background, which include conflicting word *"ripped jeans"* to invoke mode conflict. In this example, mode conflict happens to both ControlNet [143], IP-Adapter [136], and our proposed ViscoNet at a control strength of 80%, as observed with the realistic person and background in Figure 5.5e. However, our method has quickly escaped mode conflict at a control strength of 60% (Figure 5.5d), as at this point, the visual conditioning is still effective, maintaining the overall clothing styles and colours. We can also observe the harmonised transition towards the desired image style as reduced control strength tips the balance towards text prompt depicting *"Picasso"*(Figure 5.5a). Both reference methods only managed to escape mode conflict at around 40% (Figure 5.5b), which has considerably weakened pose or visual control for ControlNet and IP-Adapter, respectively.

We confirm our qualitative observation with quantitative results. Like [136], we measure the effectiveness in generating the correct image styles by employing the CLIP similarity score between the text prompt and the generated image. A high CLIP score indicates a low or absence of mode conflict. We measure control effectiveness by measuring pose accuracy using the Object Keypoint Similarity (OKS) standard in MSCOCO challenge[66]. We will also introduce new metrics to measure and interpret mode conflict better. In this experiment, we selected 5 image styles - Picasso, Van Gogh, oil painting, Ukiyoe, and stained glass that are more likely to conflict with modern clothing. We generated 20 samples at each control strength (over 5000 images). The results are listed in Table 5.1 and plotted in graph as shown in are plotted in Figure 5.6.

Strength	0.0	0.2	0.3	0.4	0.5	0.6	0.8	1.0
				CLIP score				
ControlNet	0.2720	0.2620	0.2440	0.2340	0.2300	0.2300	**0.2240**	**0.2260**
IP-Adapter	**0.2900**	0.2920	0.2780	0.2620	0.2360	0.2120	0.1780	0.1900
ViscoNet(Ours)	0.2860	**0.2940**	**0.2920**	**0.2900**	**0.2800**	**0.2660**	**0.2420**	0.2220
				CLIP accuracy				
ControlNet	0.8660	0.7720	0.7000	0.6180	0.4620	0.5500	0.5020	**0.5760**
IP-Adapter	0.9800	0.9700	0.8800	0.7500	0.6300	0.4100	0.1500	0.2100
ViscoNet(Ours)	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9000**	**0.7000**	**0.5760**
				Pose accuracy (OKS)				
ControlNet	0.0880	0.4139	**0.6610**	**0.8305**	**0.8596**	**0.8852**	**0.9223**	**0.9348**
IP-Adapter	**0.5379**	**0.5412**	0.6060	0.6813	0.7546	0.8074	0.9010	0.9298
ViscoNet(Ours)	0.0446	0.1654	0.3869	0.6580	0.7845	0.8253	0.8824	0.9102

Table 5.1: Reduced control strength results in higher CLIP scores and accuracy, translating to less mode conflict.

At 100% control strength, IP-Adapter lost most of its text capability, including changing image style (Figure 5.2e), resulting in the lowest CLIP scores (Figure 5.6a), indicating substantial mode conflict. The CLIP score for ControlNet remains constant in regions above 40%, whereas our method exhibits linear improvement in the same range. Both visual adapters effectively reduce mode conflict by using weaker control strength as indicated by weaker pose accuracy (Figure 5.6c). Our method consistently outperforms IP-Adapter in CLIP score at every control strength. It is worth noting that the IP-Adapter maintains its pose control for control strength <40% as they use separate adapters for pose control. Their

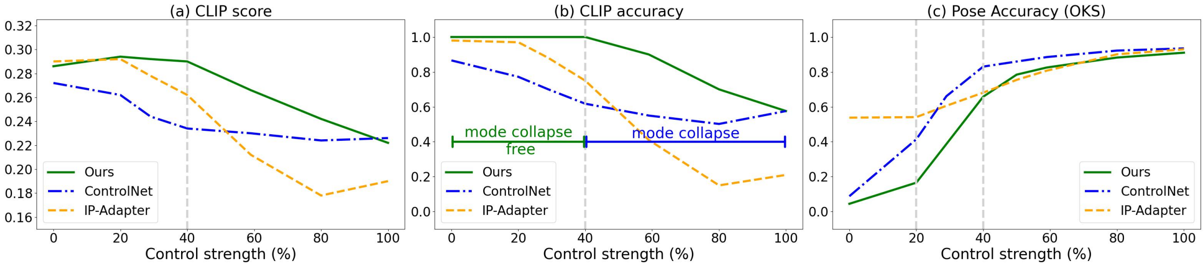


Figure 5.6: (a) Reducing control strength alleviates mode conflict, our method can escape mode conflict faster, retaining better pose and visual control (b) CLIP accuracy provides better interpretability of mode conflict (c) Level of control as measured by pose accuracy. Visual prompting methods have slightly weaker pose control.

drop of <40% is attributed to inaccuracy in pose detectors' recognition of humans in artistic painting. Our quantitative results align with qualitative observations, establishing our method's superior interpolation capability and ability to minimise mode conflict.

While CLIP scores are effective, their limitation lies in the lack of interpretability regarding the degree of mode conflict. Additionally, the absence of a standardised CLIP model within the machine learning community introduces variability, making cross-model comparisons challenging. Given these challenges, we explore alternative metrics for a more comprehensive evaluation. As mode conflict is an inherent discrete state, we employ CLIP binary classification ($CLIP_{acc}$) by comparing CLIP image embedding to two CLIP text classes - [image style],"*real photo*". More generally, two modes are compared - target mode and stuck mode. In other words, we detect mode conflict if the image is classified as a real photo when it was supposed to be in the target image style. As shown in Figure 5.6b, $CLIP_{acc}$ correlates well to the CLIP similarity score but provides a normalised score easier for interpretation and enhanced robustness against CLIP model variation. We define **mode conflict rate (MCR)** as :

$$\mathcal{MCR} := 1 - CLIP_{acc} \tag{5.3}$$

Mode conflict is a phenomenon that occurs randomly, depending on the prompts and random seeds applied. Consequently, the MCR is a batch statistic that reflects the overall method performance. In Figure 5.6b, we achieve mode conflict free at a control strength of 40% (MCR=0% or $CLIP_{acc} = 100\%$) while IP-Adapter reaches that state later at much-weakened control strength. Our method produces a higher CLIP score than the baseline at various control strengths, indicating less mode conflict. This is more evident in CLIP accuracy; at control strength 0.5, we achieve 100% (or 0% MCR) while baselines have only 46% and 63% ControlNet and IP-Adapter, respectively.

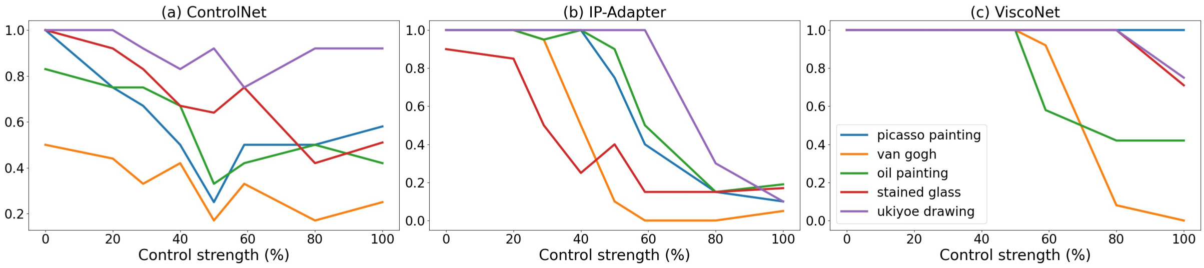


Figure 5.7: CLIP accuracy - comparing different image styles.

Figure 5.7 shows the breakdown of CLIP accuracy across the image styles in Table 5.1. Based on the same Stable Diffusion model, all models have shown the highest mode conflict rate in Van Gogh's painting style, while Ukiyoe is the least affected. Our method consistently outperform baseline models across all image styles, escaping mode conflict at higher control strength.

5.4.2 Re-identification

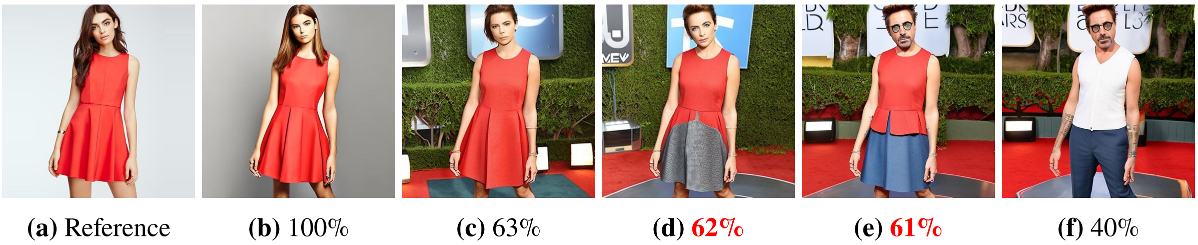

(a) Reference **(b)** 100% **(c)** 63% **(d)** 62% **(e)** 61% **(f)** 40%

Figure 5.8: IP-Adapter showing the transition from the reference image to text prompt "Robert Downey Jr." by reducing control strength. There exists a big domain gap between (d) and (e).

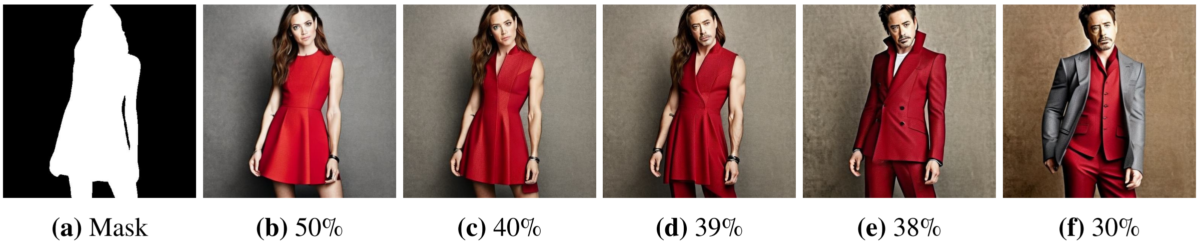

(a) Mask **(b)** 50% **(c)** 40% **(d)** 39% **(e)** 38% **(f)** 30%

Figure 5.9: Method 1 - with head mask, smoother transition with smaller mode gap between (c) and (d).

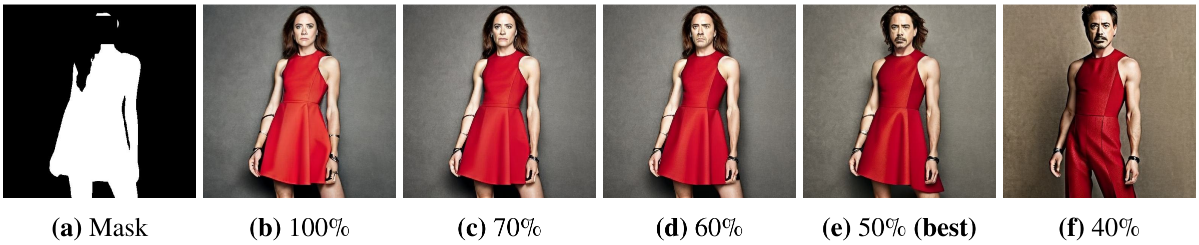

(a) Mask **(b)** 100% **(c)** 70% **(d)** 60% **(e)** 50% **(best)** **(f)** 40%

Figure 5.10: Method 2 - without head mask. Smooth transition with (e) achieving good balance to deliver the desired result. The face and hair mask are detected and removed by segmentation tool. Our method has good tolerance over the mask region and does not require it to be pixel-accurate.

We formulated a demanding task to scrutinise an extreme instance of domain gap and assess the qualitative efficacy of visual prompting methods in addressing such challenges. In this task, the goal is to transform the person's identity in the reference image into the person depicted in the text prompt, all while preserving the original clothing depicted in the reference image. In Figure 5.8, we show that decreasing control strength in IP-Adapter morphs the face towards the target (Robert Downey Jr.) at the expense of clothing faithfulness (red dress). A small control strength change between Figure 5.8d and 5.8e causes a significant shift in the image, indicating a big domain gap it fails to bridge harmoniously.

This common problem also affects our default configuration Method 1, which uses full human tasks. It achieves good results close to the target as shown in Figure 5.9d. Through extensive experimentation, we discovered that the face has disproportionately influenced the entire image generation process. Consequently, it becomes imperative to substantially reduce the control strength (to around 40% in this example) to mitigate the impact of the face, albeit at the expense of visual control. Leveraging our novel architecture, we can effectively bridge this gap by selectively excluding the face from the feature mask, as shown in Figure 5.10 (Method 2). In essence, this action prevents the control signal from reaching the face region of the LDM. We tried applying a similar approach to the IP-Adapter by masking off the head from the reference image in pixel space, but this proved ineffective. This underscores the efficacy of our novel architecture in harmonising text-visual controls. This has also proved useful in escaping mode conflict in certain challenging styles in stylisation tasks.

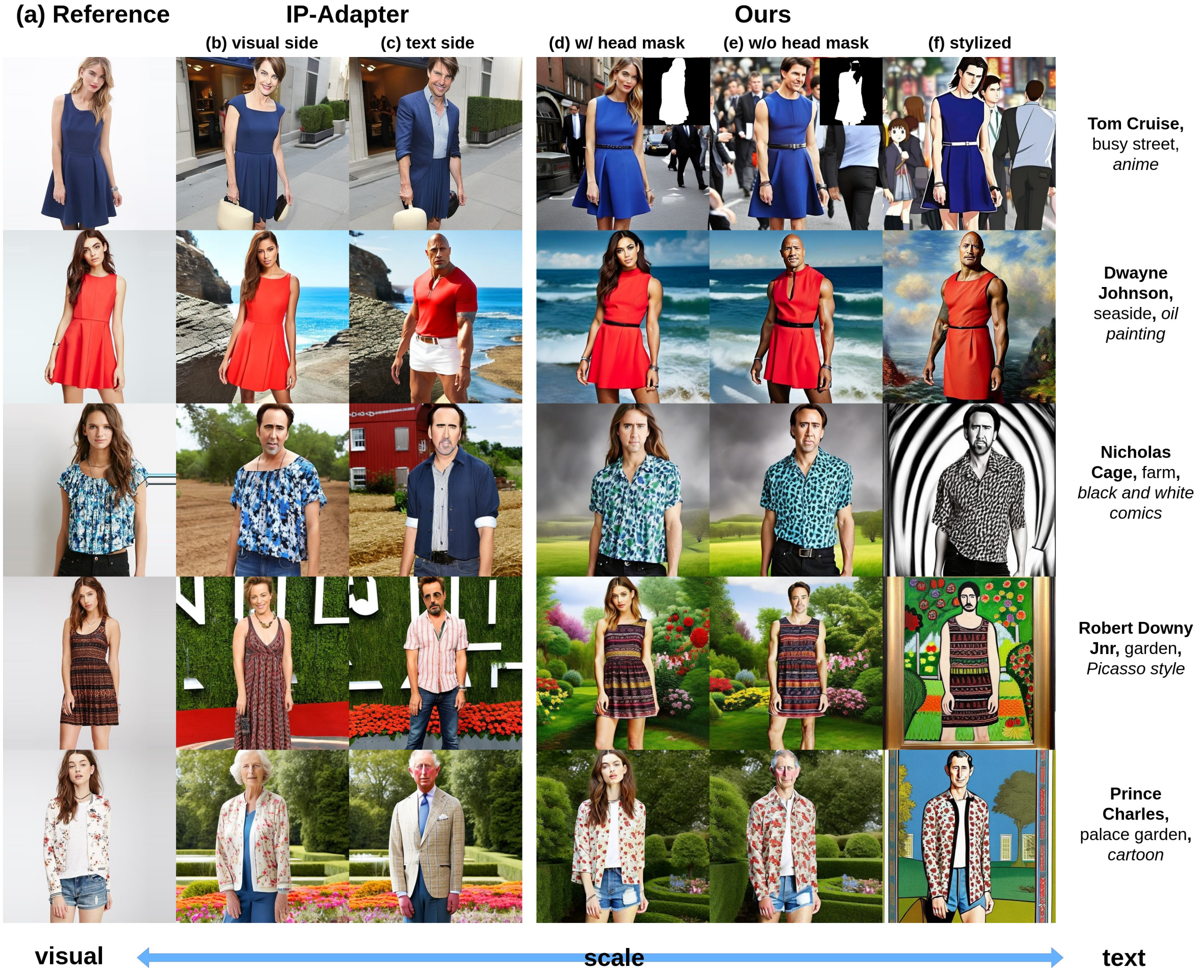

(a) Reference
IP-Adapter
(b) visual side
(c) text side
Ours
(d) w/ head mask
(e) w/o head mask
(f) stylized
Tom Cruise, busy street, anime
Dwayne Johnson, seaside, oil painting
Nicholas Cage, farm, black and white comics
Robert Downy Jnr, garden, Picasso style
Prince Charles, palace garden, cartoon
visual
scale
text

Figure 5.11: Challenging re-identification task to transform female in (a) reference image to male celebrities depicted in text prompt. We included an additional stylisation step (not included in the result) to demonstrate our ability to bridge the domain gap.

We generated over 7,560 images per method to perform a detailed quantitative study, covering a broad range of variations to assess control accuracy and visual quality. Our dataset included seven distinct male celebrity names in the text prompts, six reference clothing items, and nine control strength levels, with each control level generating 10 samples. This comprehensive sampling strategy enabled us to evaluate how varying control strengths influenced the generated images' fidelity to both textual and visual conditioning. Some test samples (input image and text) are shown in Figure 5.11. We include the image background and style to demonstrate our capability to maintain a constant background and bridging domain gaps across multiple dimensions to achieve stylisation. We do not include them in our experiments as the objective is the foreground person identity and clothing. The experiment results (Table 5.2) are summarised in Figure 5.12. The presence of steep change in CLIP score (Figure 5.12a)

Strength	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.8	1.0
					CLIP score				
IP-Adapter	0.3066	0.3052	0.3003	0.2910	0.2729	0.2546	0.2231	0.1687	0.1606
Method 1 (ours)	0.3223	0.3230	0.3218	0.2993	0.2139	0.1846	0.1768	0.1670	0.1628
Method 2 (ours)	**0.3232**	**0.3254**	**0.3250**	**0.3218**	**0.3196**	**0.3064**	**0.2920**	**0.2544**	**0.2312**
					Mode Conflict Rate (MCR)				
IP-Adapter	0.05	0.06	0.11	0.19	0.32	0.45	0.67	0.99	1.00
Method 1 (ours)	**0.00**	**0.00**	**0.00**	0.12	0.74	0.97	1.00	1.00	1.00
Method 2 (ours)	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.08**	**0.15**	**0.40**	**0.57**
					MS-SSIM				
IP-Adapter	0.1248	0.1420	0.1705	0.2033	0.2452	0.2888	0.3700	0.4447	0.5107
Method 1 (ours)	**0.1514**	**0.1795**	**0.2312**	**0.3781**	**0.5043**	**0.5335**	**0.5498**	**0.5536**	**0.5409**
Method 2 (ours)	0.1507	0.1627	0.1886	0.2538	0.3661	0.4387	0.4742	0.5173	0.5183

Table 5.2: Reduced control strength results in higher CLIP scores and accuracy, translating to less mode conflict.

and MCR (Figure 5.12b) with our Method 1 proves the evident domain gap within the 30%-50% control strength range. However, removing the face from the mask in Method 2 drastically improves performance, outperforming IP-Adapter considerably. On the other hand, we measure effectiveness of visual control with image similarity score MS-SSIM [126] (Figure 5.12c). Method 1 (and 2) is consistently higher than IP-Adapter in MS-SSIM, suggesting more faithful visual appearance once escaping mode conflict (Figure 5.8 and 5.9) .

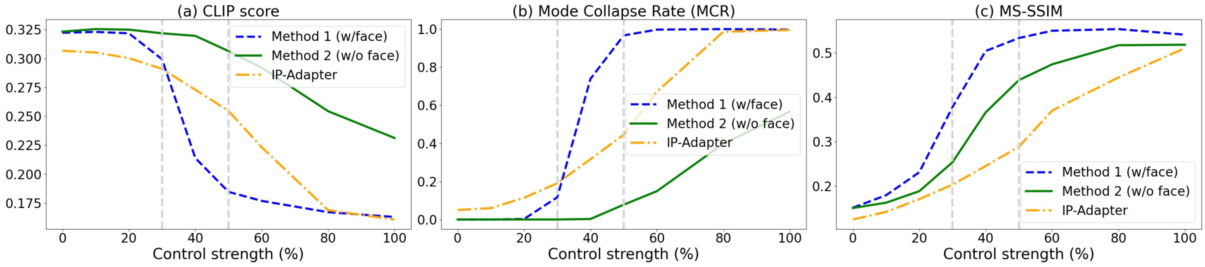
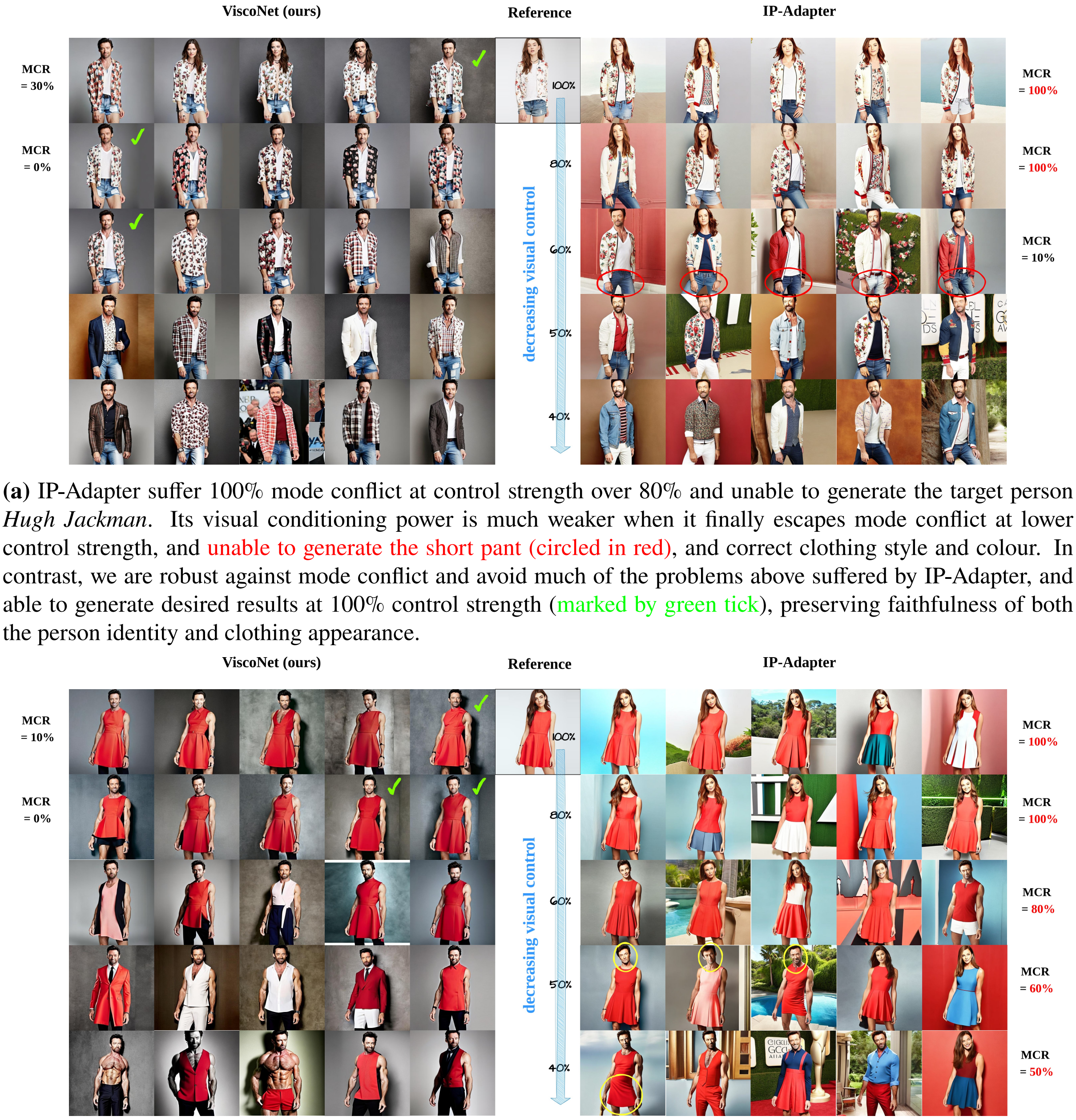


Figure 5.12: Quantitative result showing the effectiveness of our method to escape mode conflict in the challenging re-identification task.

5.4.3 Control Strength Analysis

In Figure 5.13, we show experiment samples of text prompt *Hugh Jackman* and reference image 2, where we randomly sample 50% of the samples for various control strengths from both our and IP-Adapter. With 100% strength, although IP-Adapter can reconstruct the reference image, it suffers 100% MCR (Mode Conflict Rate) of that particular control strength, reference image, and text prompt). In contrast, our method can generate the target person's face in the reference clothing. Mode conflict persists with IP-Adapter until 60% strength, at which point the clothing it generates is getting more random, and most notably, it can no longer generate the short pants from the reference image. Next, we replace the reference with the short dress as shown in 5.13b. The conflict between the feminine dress and Hugh Jackman's masculine image creates a wider domain gap, which results in persistent mode conflict, worse in the previous example. It has just begun to leave mode conflict at 50% control strength, showing strange



(a) IP-Adapter suffer 100% mode conflict at control strength over 80% and unable to generate the target person *Hugh Jackman*. Its visual conditioning power is much weaker when it finally escapes mode conflict at lower control strength, and unable to generate the short pant (circled in red), and correct clothing style and colour. In contrast, we are robust against mode conflict and avoid much of the problems above suffered by IP-Adapter, and able to generate desired results at 100% control strength (marked by green tick), preserving faithfulness of both the person identity and clothing appearance.

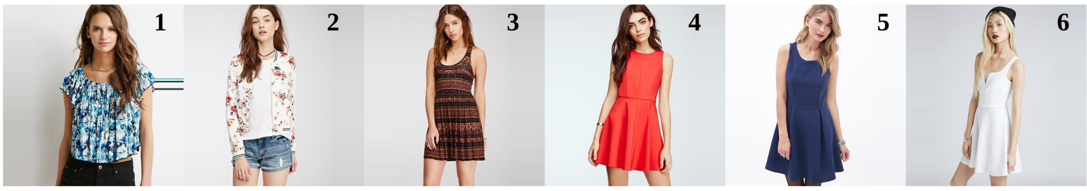
(b) The conflict between the feminine reference image and Hugh Jackman's masculine image creates more conflict and hence mode conflict as suffered by IP-Adatper. IP-Adatper struggles to generate correct faces and pleated dress patterns (circled in yellow) at weaker control strength. This does not affect our method.

Figure 5.13: Comparing the effect of control strength on re-identification task. IP-Adapter suffers much more severe mode conflict and struggles to create perfect image balancing the reference image and text prompt of *Hugh Jackman*.

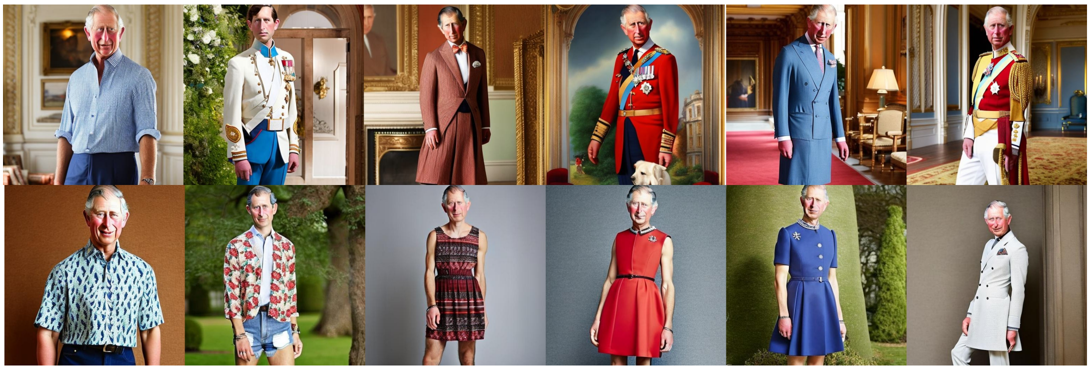
faces that don't quite look like the movie star. When it finally escapes mode conflict at 40%, it has lost the visual details of the pleated pattern on the dress (circled in yellow).

Overall, our method is effectively mode conflict free at 60% while IP-Adapter still 5.2 has 67% MCR. As shown in Figure 5.13, although high control strength introduces some mode conflict to our method. However, we can still generate high-quality images, preserving visual conditioning and a person's identity.

We further explore qualitative results in this section. Unlike Figure 5.8- 5.11, where we slide along the control strength on the same random seed to demonstrate latent space discontinuity, we extend the experiment to present the best samples across all control strengths from both methods for direct comparison, as shown in Figure 5.14-5.16.



(a) Visual reference taken from the unseen test dataset.

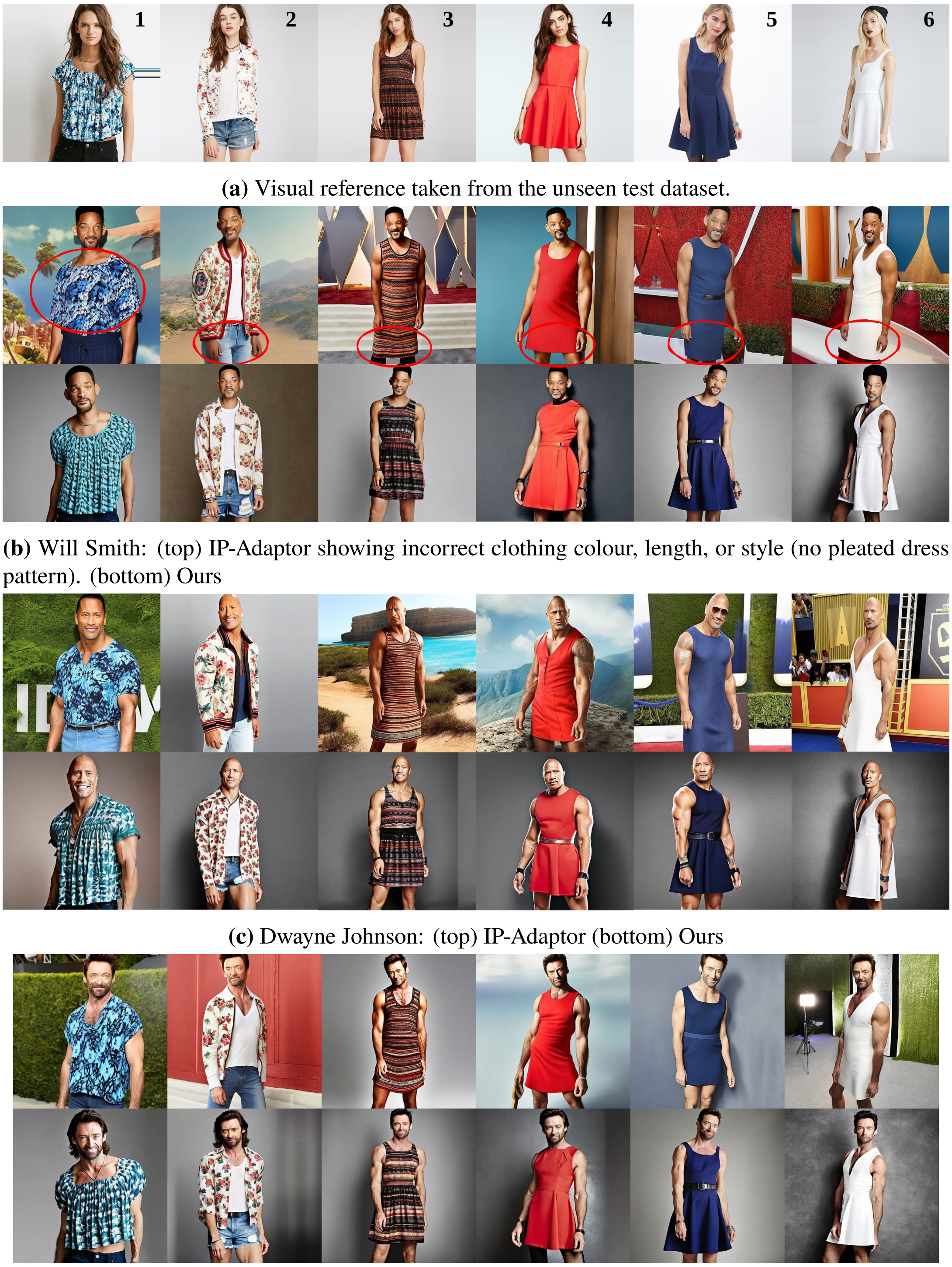


(b) Unlike other movie stars with more diverse costumes, Prince Charles' limited clothing range presents the toughest challenge. (Top) IP-Adatper cannot produce any image of Prince Charles wearing the reference clothing. (Bottom) despite the extreme data gap, our method can produce reasonable images.

Figure 5.14: Most challenging example in re-identification task - Prince Charles.

Among all the celebrities mentioned in the text prompt, *Prince Charles*[1] - known for having a limited wardrobe of formal attires in public images - presents the greatest challenge to the generalisation capability of the models. IP-Adapter encounters difficulties and fails to generate any image of Prince Charles in casual or feminine clothing, as depicted in the reference image (Figure 5.14). In contrast, our method achieves reasonable success despite the monumental challenge. Figure 5.15 - 5.16 shows samples from the rest of the text prompts used in the experiment. Overall, IP-Adapter needs to have much-lowered control strength to escape mode conflict, resulting in loss of fidelity in clothing to the reference images, including the incorrect length of pants or dress, wrong colour and pattern, i.e., loss of the pleated dress pattern, it previously able to generate (Figure 5.13b).

[1] Stable Diffusion was trained on dated data before Prince Charles ascended to be king, so we adhere to his old title in the experiment.



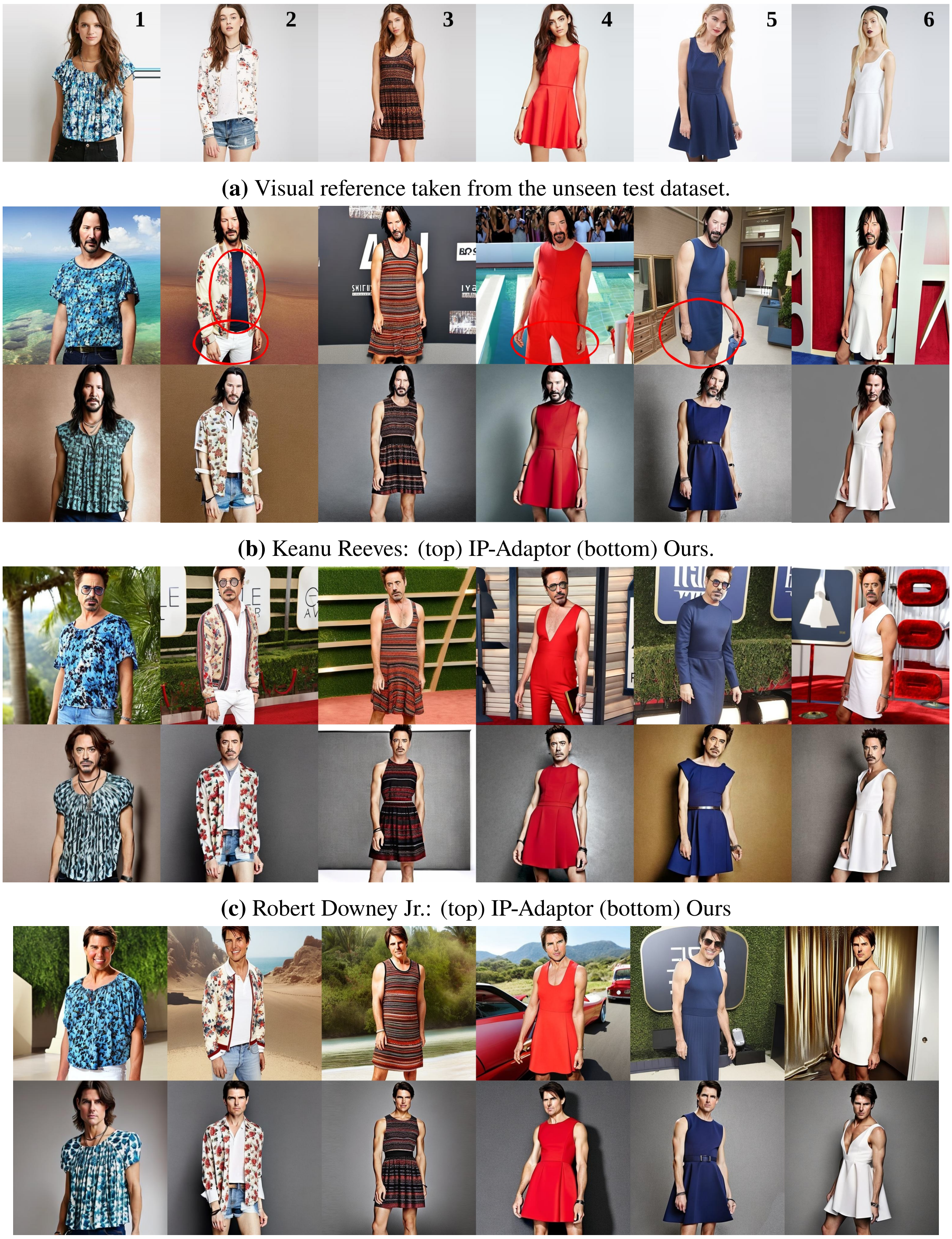
(a) Visual reference taken from the unseen test dataset.

(b) Will Smith: (top) IP-Adaptor showing incorrect clothing colour, length, or style (no pleated dress pattern). (bottom) Ours

(c) Dwayne Johnson: (top) IP-Adaptor (bottom) Ours

(d) Hugh Jackman: (top) IP-Adaptor (bottom) Ours.

Figure 5.15: Re-identification comparison with IP-Adapter.



(a) Visual reference taken from the unseen test dataset.

(b) Keanu Reeves: (top) IP-Adaptor (bottom) Ours.

(c) Robert Downey Jr.: (top) IP-Adaptor (bottom) Ours

(d) Tom Cruise: (top) IP-Adaptor (bottom) Ours

Figure 5.16: Re-identification comparison with IP-Adapter.

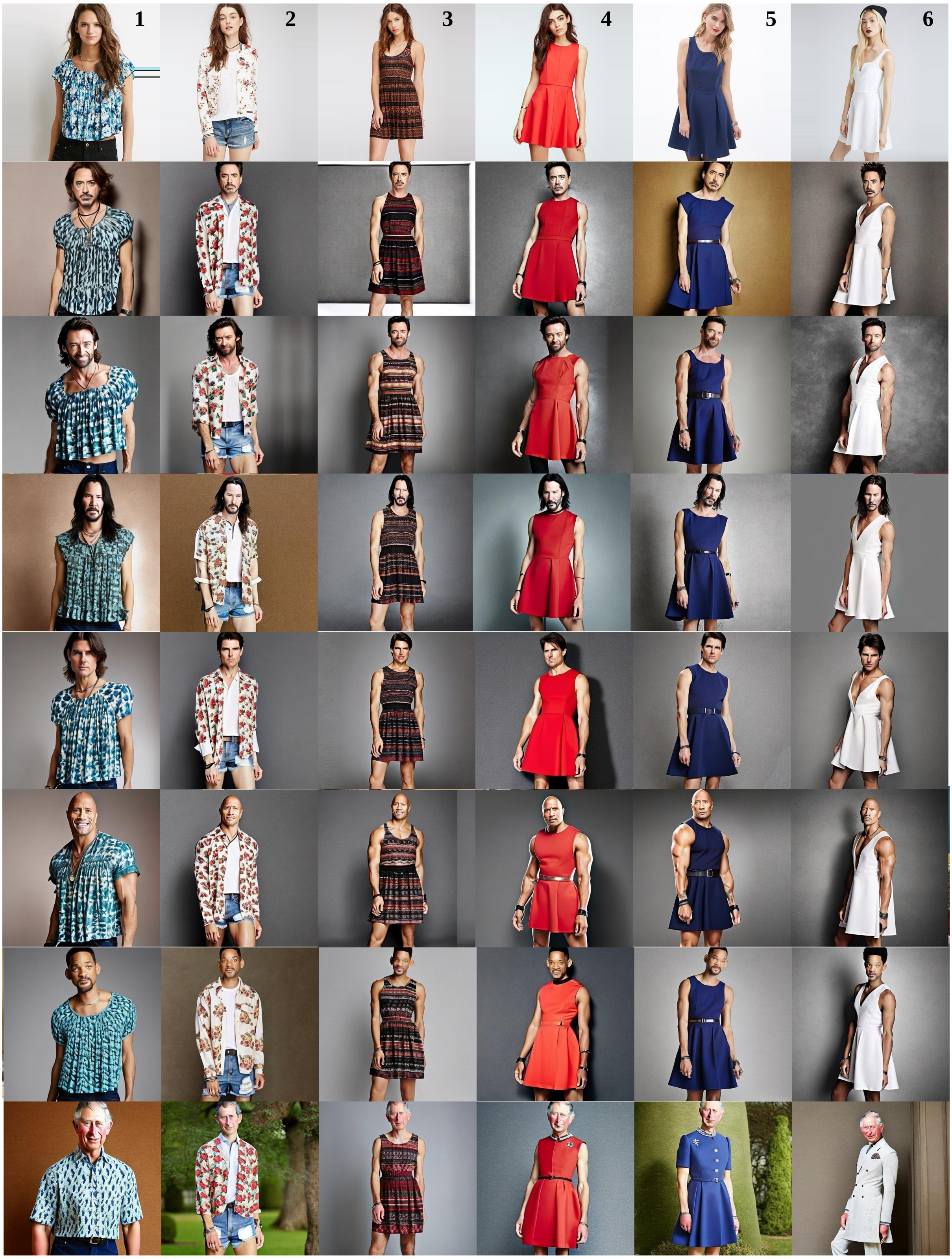
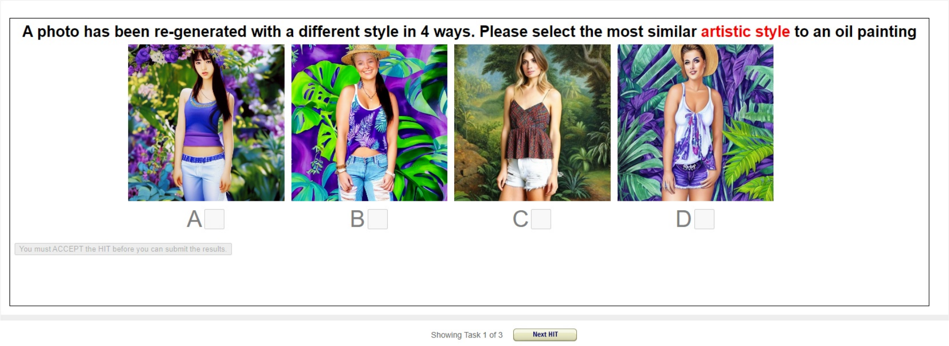


Figure 5.17: Putting our images together shows the consistency of our method in delivering celebrity re-identification.

5.4.4 Generating Diverse Human Image Styles

Lastly, we also performed large-scale human evaluation comparing specialist SOTA pose-guided HIG models HumandSD [47], ControlNet [143], and T2I-Adapter [77]. In this experiment, we generate 1400 images evenly across seven image styles and the models (Figure 5.19). We use text prompts to describe clothing for reference methods and visual prompts for our process. In each test sample, 221 human evaluators were randomly shown a sample from each model and asked to pick one that best matches the text prompt as shown in Figure 5.18. . The majority, 55% of 700 responses (Table 5.3), prefer our samples, proving overall superiority in image quality and visual control.

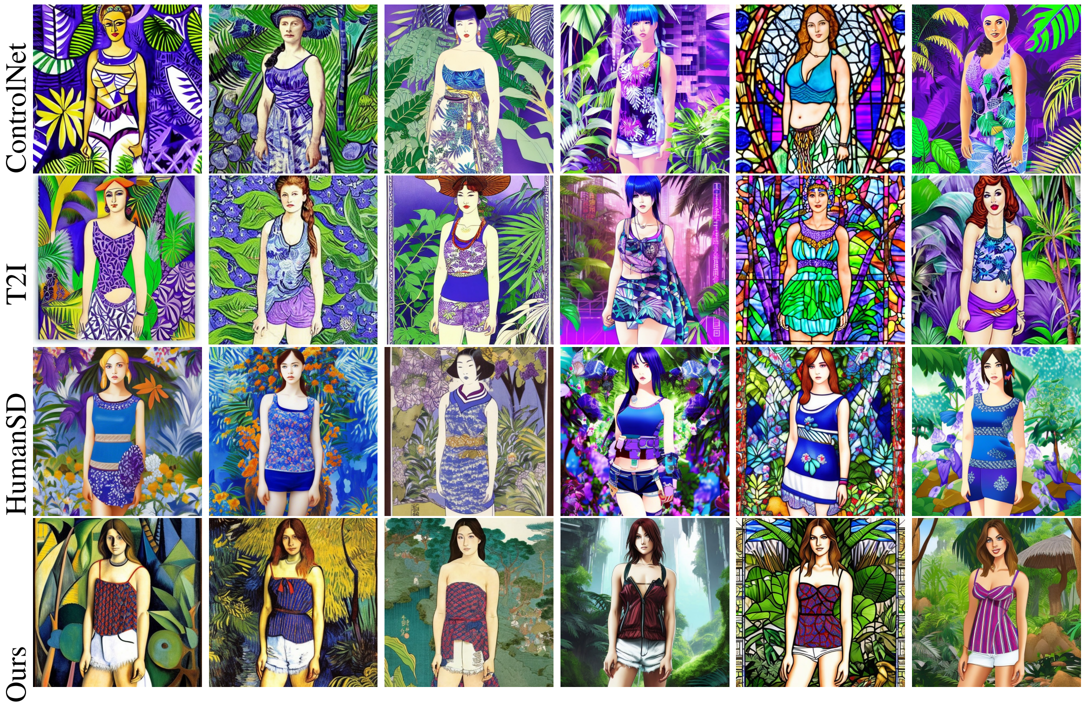


A photo has been re-generated with a different style in 4 ways. Please select the most similar artistic style to an oil painting

You must ACCEPT the HIT before you can submit the results.

Showing Task 1 of 3

Figure 5.18: Screenshot of user study presented to users for evaluating the quality of the stylisation against the three baselines.



(a) Picasso **(b)** Van Gogh **(c)** Ukiyoe **(d)** cyberpunk **(e)** stained glass **(f)** Disney

Figure 5.19: All reference methods have the purple clothing colour spread into the forest background, while our method avoids this problem and can generate a vibrant and diverse background.

	Human Evaluation				
Image Styles	HumanSD	ControlNet	T2I-Adapter	**ViscoNet (Ours)**	Ours (%)
Ukiyoe	27	32	4	**37**	**37%**
Cyberpunk anime	23	13	21	**41**	**41%**
Stained glass	0	32	23	**45**	**45%**
Van Gogh	2	13	9	**76**	**76%**
Picasso	0	13	42	**45**	**45%**
Oil Painting	9	11	7	**73**	**73%**
Disney	5	23	5	**67**	**67%**
Total	77	139	111	**384**	
Average	9.43%	19.9%	15.9%	**54.9%**	

Table 5.3: Our method scores the highest in human evaluation, proving its ability to generate good-quality, diverse image styles.

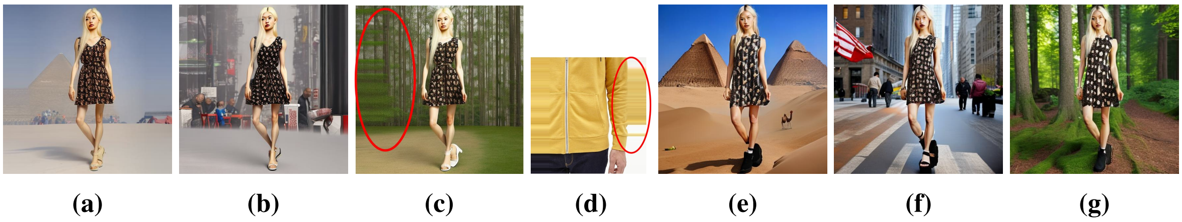


Figure 5.20: Without feature masking in training :(a)-(c) pale, dull background (d) padding artifact from dataset. Using feature masking in training: (e)-(g) vibrant colours and rich background.

5.5 Ablations

Necessity of Feature Mask in Training. Catastrophic forgetting can be demonstrated using the DeepFashion dataset; in our initial experiments, we applied feature masking to the training loss function but excluded it from the control signals. However, applying the feature mask post-training is ineffective, as shown by the pale and dull background in Figure 5.20a -5.20c. In particular, the artifact (circled in red) in Figure 5.20c gives a clear indication of leakage of background originating from padding artifact uniquely caused by our dataset pre-processing error as shown in Figure 5.20d. Evidently, our method of applying feature masking in training produces a vibrant background (Figure 5.20e-5.20g), demonstrating that our method is effective in avoiding catastrophic forgetting.

CLIP Local Image Embedding Captures Fine Texture. We experimented with two image embedding methods for visual conditioning - global and local CLIP image embedding. Figure 5.21 shows that local CLIP embedding used in our method is better at capturing fine texture details.

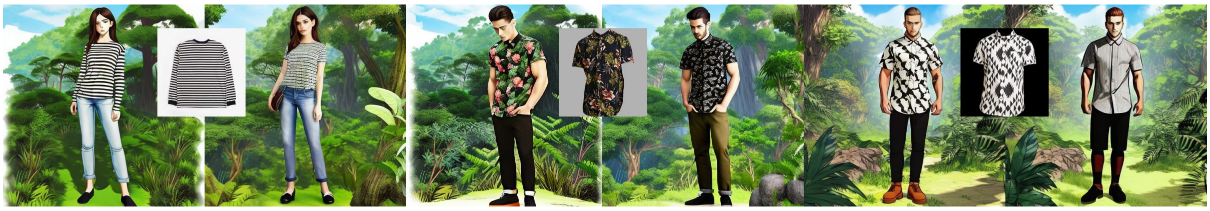


Figure 5.21: Local CLIP image embedding used in our method can capture fine texture details. (left) local embedding (mid) visual prompt (right) global embedding.

5.6 Versatile Human Image Generation Tasks

Our method, which integrates multiple input modalities as conditioning controls, enhances both controllability and flexibility, enabling a range of human image generation tasks with notable versatility. This approach allows for nuanced manipulations, such as person re-identification using image or text prompts, image stylisation, pose retargeting, and virtual try-on applications. The flexibility of multimodal input conditions helps to achieve a higher degree of precision and variety in the generated outputs, catering to diverse user intents. In the following section, we present examples generated using this approach, illustrating the effectiveness of multimodal conditioning in producing controlled, photorealistic human images under various scenarios.

5.6.1 Re-identification (visual prompt)

Figure 5.22 shows by conditioning on face and hair images, our method generates realistic people with diverse skin tones and body shapes correctly matching the faces despite the DeepFashion dataset consisting of more than 90% of female images, predominately fair-skinned women.

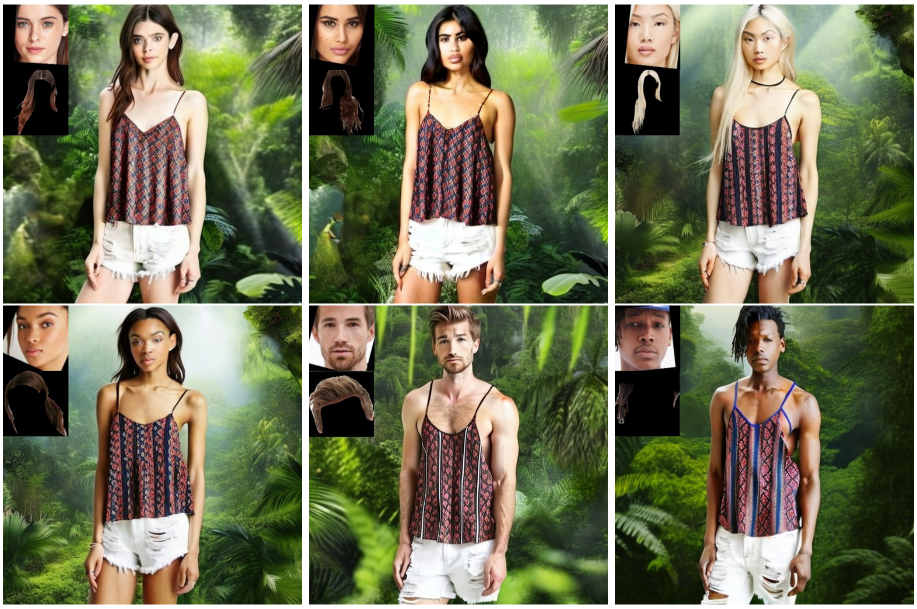
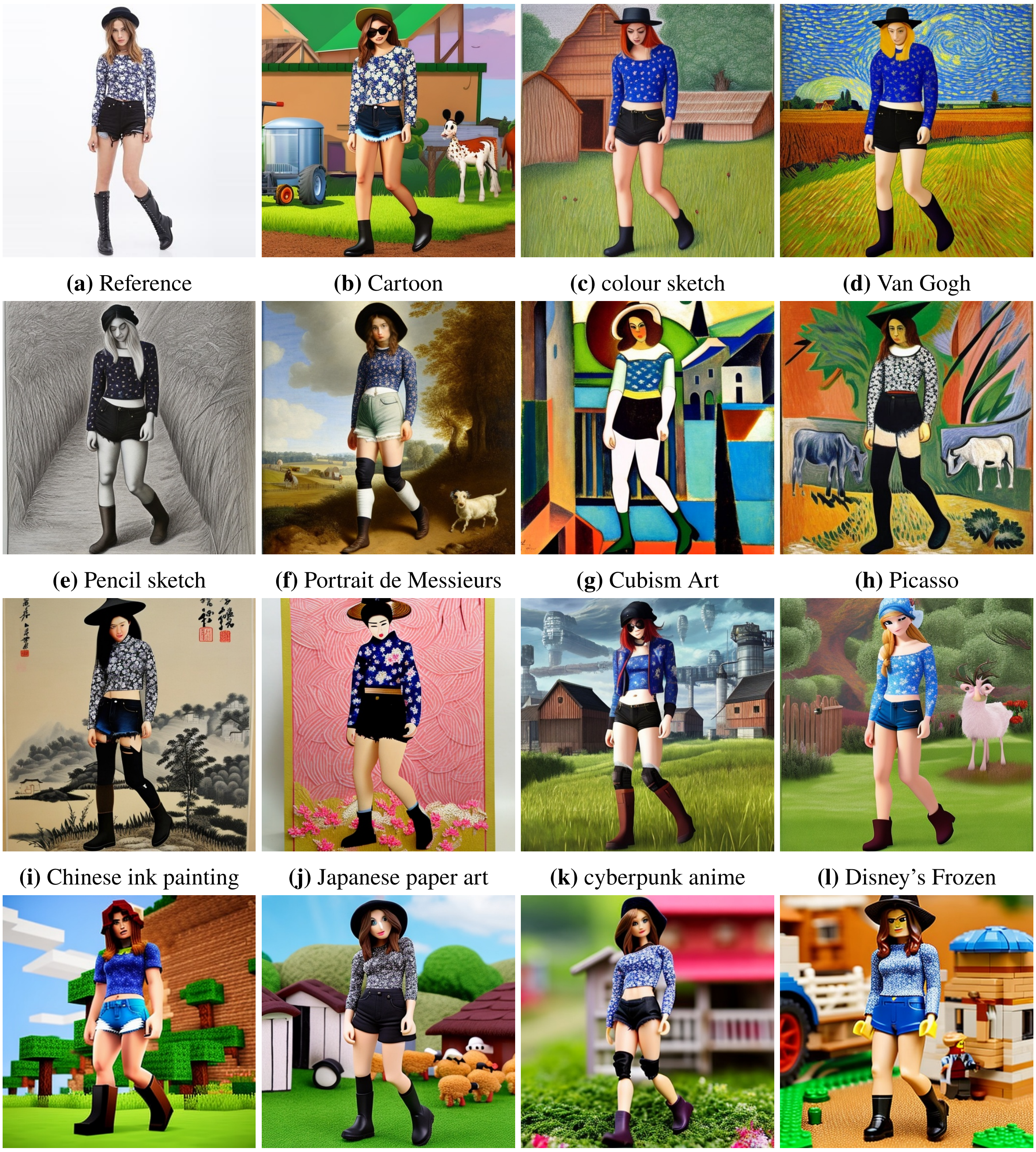


Figure 5.22: Re-identification with a visual prompt.

5.6.2 Stylisation

Figure 5.23 and Figure 5.24 show that our visual conditioning is effective across many image domains in creating a desired person's appearance, including various painting styles and also 3D objects such as statues, sculptures, toys, and 3D graphics. Some image domains have distinctive characteristics with considerable divergence from real photos, such as cartoons with disproportionate bigger heads, which can lead to a higher mode conflict rate. We circumvent this by removing the face mask to create results such as in Figure 5.23l and Figure 5.24l.



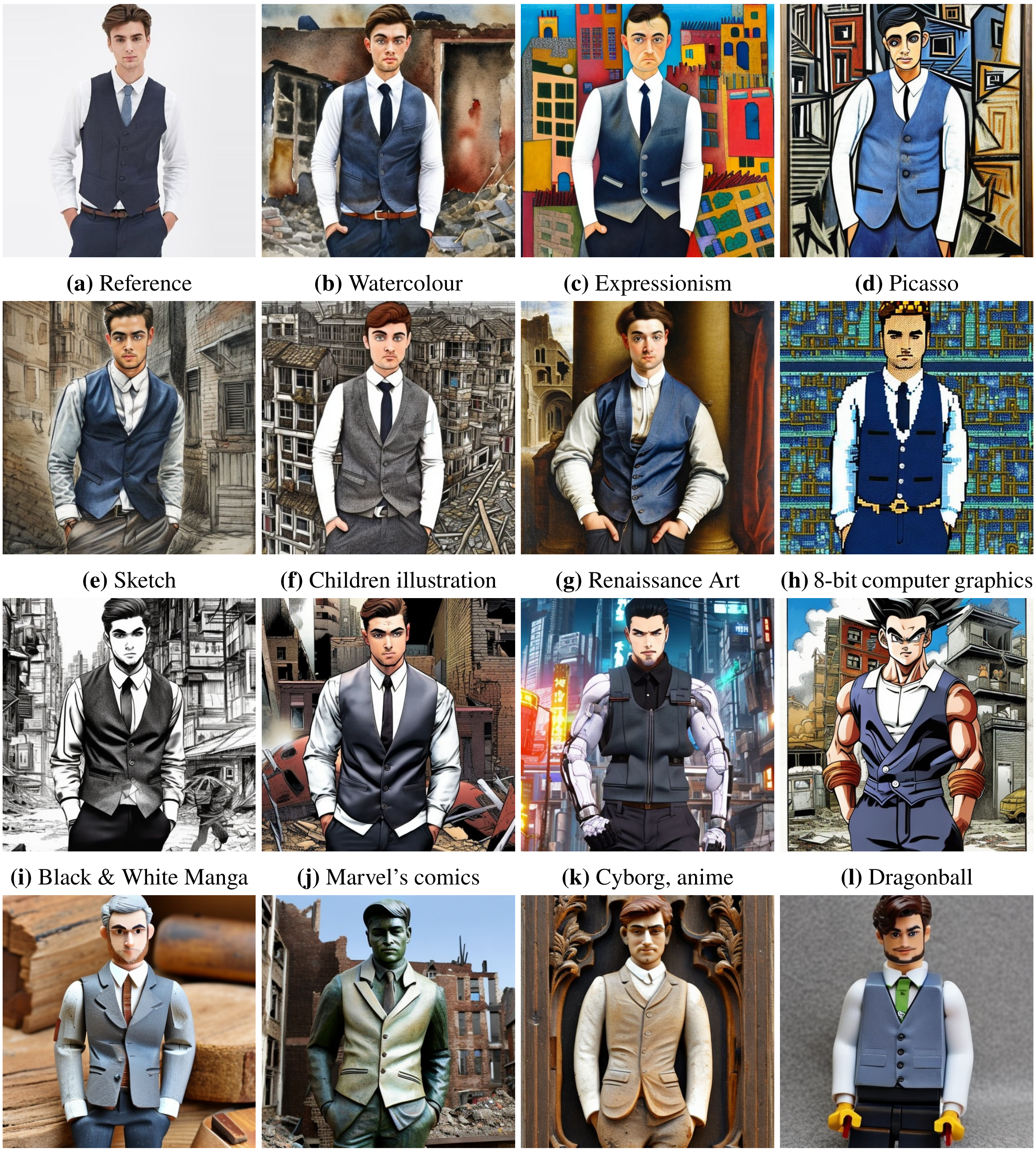
(a) Reference **(b)** Cartoon **(c)** colour sketch **(d)** Van Gogh

(e) Pencil sketch **(f)** Portrait de Messieurs **(g)** Cubism Art **(h)** Picasso

(i) Chinese ink painting **(j)** Japanese paper art **(k)** cyberpunk anime **(l)** Disney's Frozen

(m) Minecraft **(n)** Shaun The Sheep **(o)** Barbie doll **(p)** Lego

Figure 5.23: stylisation. Text prompt: *"a woman, in farm."*



(a) Reference **(b)** Watercolour **(c)** Expressionism **(d)** Picasso

(e) Sketch **(f)** Children illustration **(g)** Renaissance Art **(h)** 8-bit computer graphics

(i) Black & White Manga **(j)** Marvel's comics **(k)** Cyborg, anime **(l)** Dragonball

(m) Wooden toy **(n)** Stature **(o)** Wood Carving **(p)** Lego

Figure 5.24: stylisation. Text prompt: *"a man, in a derelict city."*

5.6.3 Pose Re-target

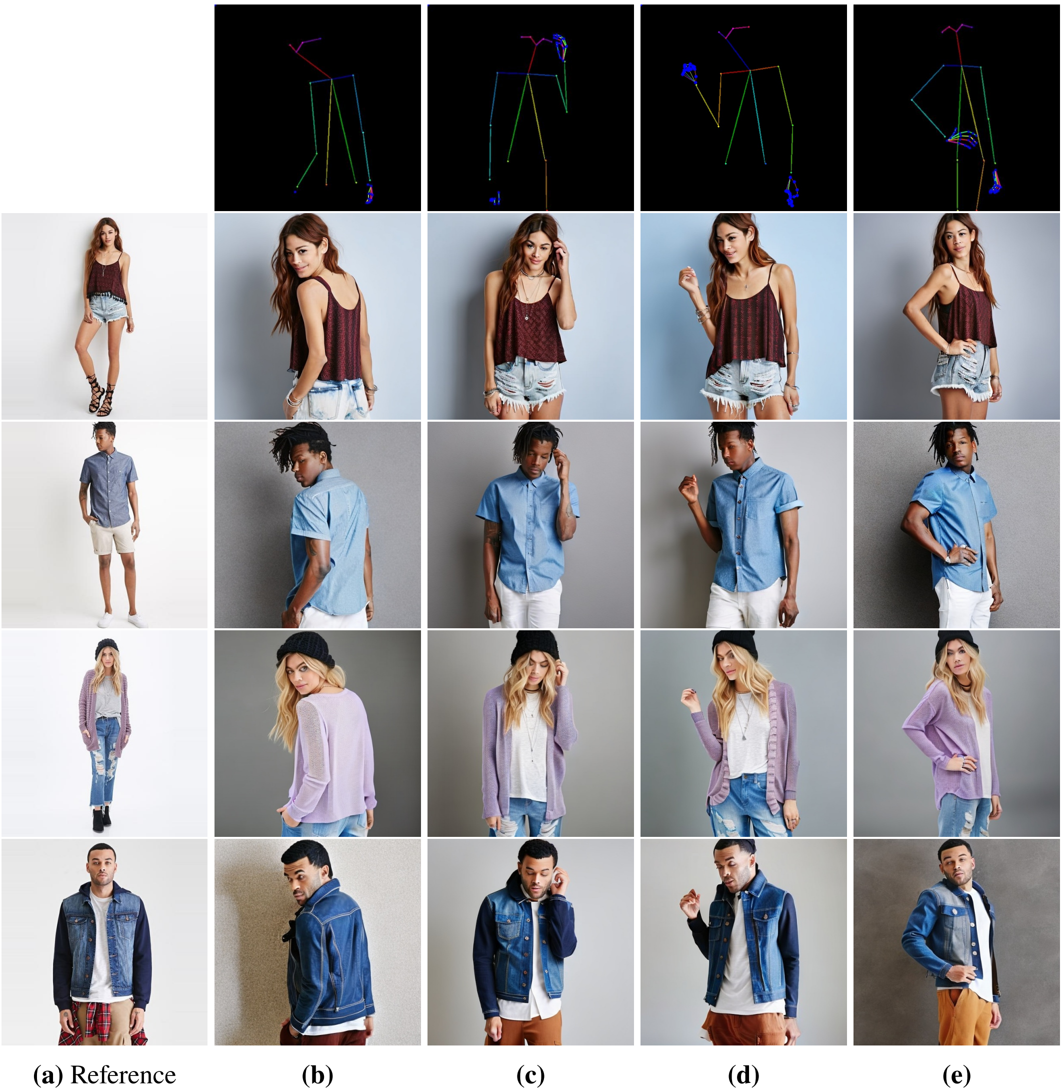
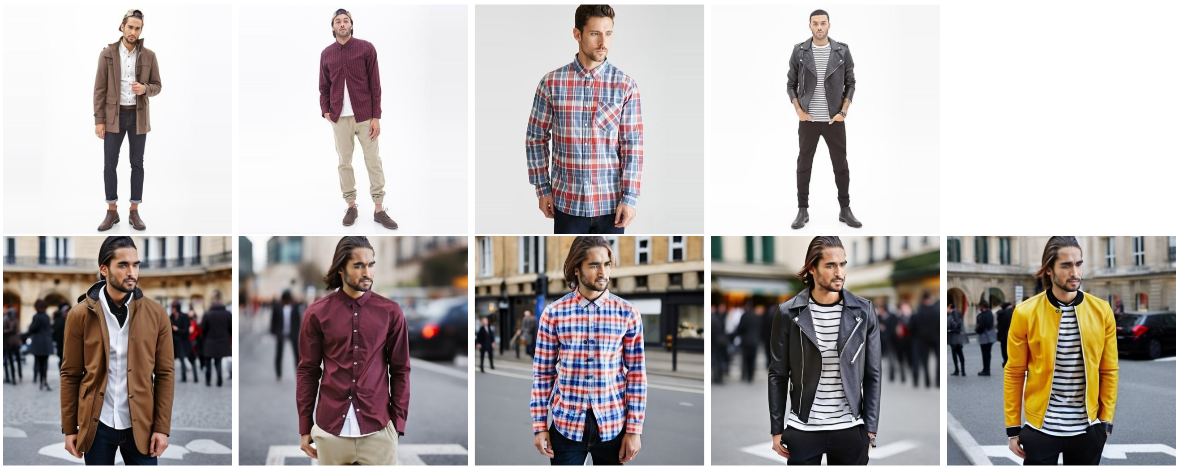


Figure 5.25: Pose Transfer from (a) reference person to new poses in (b)-(e)

5.6.4 Virtual Try-on

Figure 5.26 demonstrates how we perform fashion virtual try-on using visual and text prompts. Figure 5.27 illustrates the culmination of our methods, showcasing the seamless integration of re-identification, virtual try-on, and pose re-target.



(a) **(b)** **(c)** **(d)** **(e)** *"yellow jacket"*

Figure 5.26: High-resolution virtual try-on with real-world background. (Top) reference fashion for visual conditioning. (Bottom): virtual try-on results.

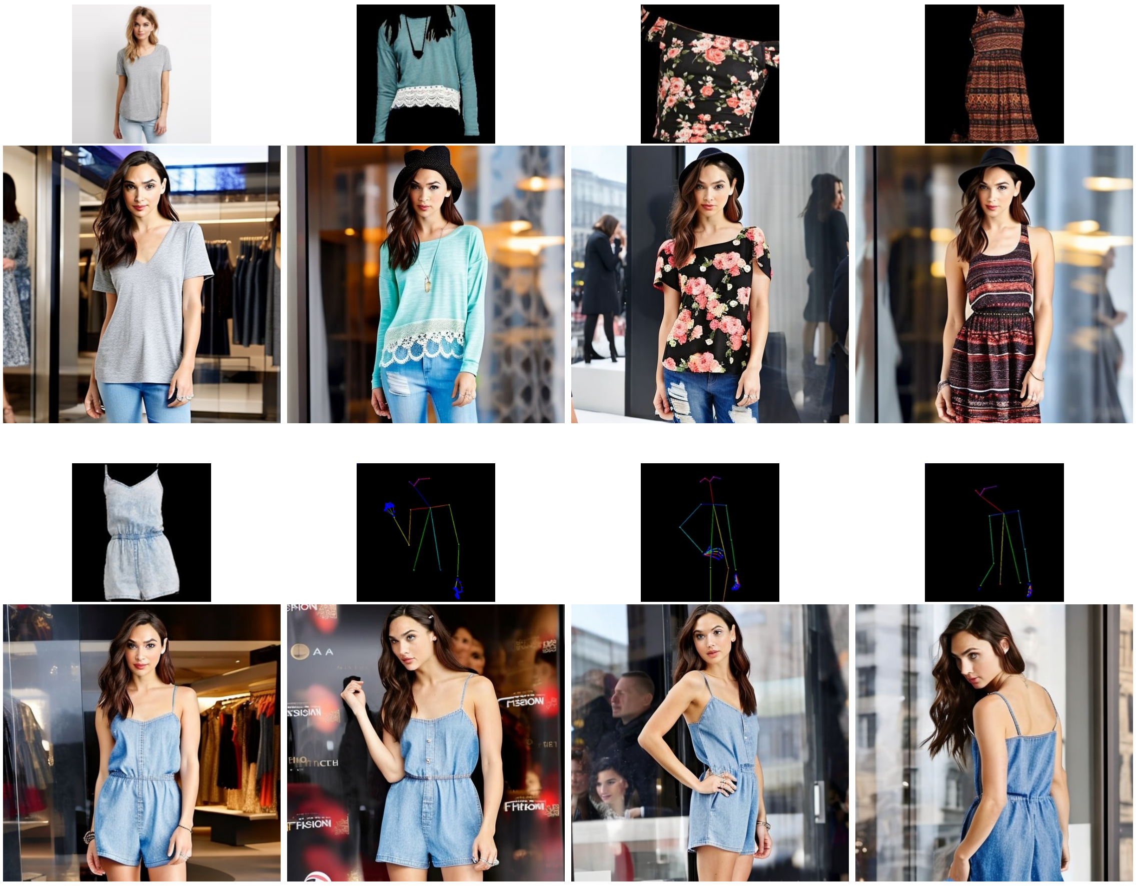


Figure 5.27: Combining re-identification, virtual try-on, and pose re-target, we showcase examples of posing fashion with celebrity avatars.

5.7 Limitations

Like IP-Adapter, the clothing colour in generated images is shaped by the inherent randomness in the initialised latent variables of the LDM backbone. While visual prompting proves effective with our method, attaining consistent and faithful image reconstruction necessitates careful selection of random seeds. On the other hand, by design choice, our model's visual prompting method learns only the foreground people and leaves the background generation to the LDM backbone. Conversely, pose transfer requires perfect reconstruction of the image background solely from the reference image. Consequently, we refrained from conducting a large-scale evaluation in virtual try-on and pose transfer tasks. Nevertheless, through careful random seed selection, we can still generate high-quality virtual try-on, pose transfer, and face swap images.

5.8 Conclusions

We present ViscoNet, a pioneering approach that seamlessly integrates visual control into a structural adapter. Our method, characterised by a single branch handling both pose and visual control stands out for its lightweight design and significantly smaller footprint when compared to existing two-adapter solutions. Through a comprehensive blend of qualitative and quantitative assessments, we demonstrate the remarkable efficacy of ViscoNet in seamlessly bridging and harmonising text and visual prompts. This unique capability not only mitigates mode conflict but also empowers the model to excel across diverse tasks, positioning it as one of the most versatile human image generation models available. Furthermore, our feature masking technique significantly contributes to our model's strength by preserving the generative power of the backbone image model. Remarkably, this is achieved despite training exclusively on a human image dataset that is orders of magnitude smaller than the datasets used by reference methods. This underscores the efficiency and generalisation prowess of ViscoNet in handling image generation tasks with limited training data.

Extending beyond static image generation, in next chapter, our focus shifts to dynamic video generation, where camera motion can be controlled with precision.

Chapter 6

Guiding Camera Motion in Video Diffusion Transformer

In Chapter 4, we demonstrated that camera pose in image generation can be controlled through straightforward parameters, such as translations along the x, y, and z axes, and rotation around the z-axis for targeted orientation adjustments. This provided a foundation for spatial control within individual images. Moving forward, this chapter expands into the more complex realm of camera conditioning specifically for video generation, on a newly emerged diffusion transformer architecture.

Supplementary video can be downloaded from project page:

`https://soon-yau.github.io/CameraMotionGuidance/`

6.1 Introduction

Recent advances in text-to-video (T2V) generation have significantly improved video quality, with diffusion models playing a key role in producing coherent and visually appealing outputs. While these models effectively translate text prompts into dynamic videos, they often lack fine-grained control over aspects like camera pose. To address this, methods such as [35, 127, 130] introduced camera pose conditioning into U-Net-based ([96]) diffusion models pretrained on 2D images ([84, 95]), demonstrating promising results in controlling camera trajectories in generated videos. Recently, transformer-based diffusion models (DiT) ([82]) have emerged as the preferred architecture for large-scale video generation models ([68, 73, 135, 148]) due to their superior scalability.

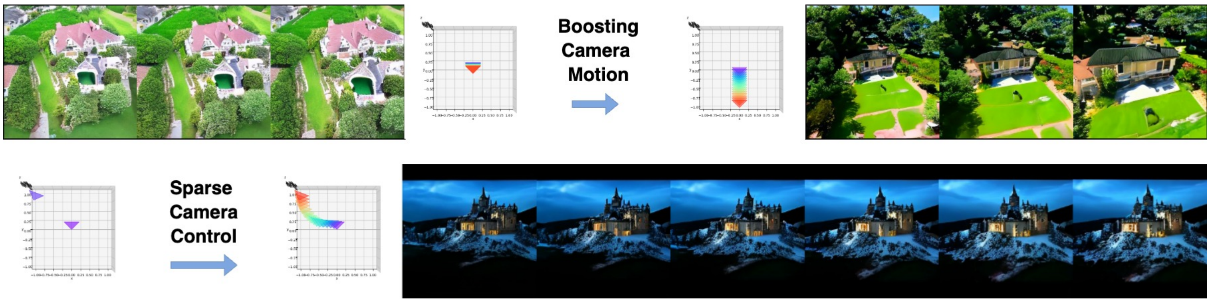


Figure 6.1: (top) The existing camera control methods implemented for DiT suffer from severe degradation in controllability and produce minimal camera motion. Our methods restore controllability and significantly boost camera motion. (bottom) Our data augmentation pipeline enables smooth camera motion even with sparse camera control, successfully interpolating between only a few or even a single provided camera pose to generate coherent and fluid camera trajectories.

However, existing camera control methods may not be effective for DiT due to the architectural differences. Concurrent work ([5]) found that direct porting of these methods to DiT led to loss of controllability, but no in-depth investigation was conducted to isolate the root causes. The introduction of new conditioning methods, alongside different camera pose representations, new DiT architecture, and the shift from space-time (2D+1D) to spatio-temporal(3D) latent encoding, further complicated the issue. To address this, we conducted an extensive study to identify the causes of camera control degradation in DiT architectures and proposed solutions broadly applicable across all DiT models and U-Net models.

Our experiments reveal that the strength of camera conditioning weakens in DiT due to the larger embedding dimensions in transformers compared to U-Net. Specifically, the 12-parameter extrinsic camera parameters, a common camera pose representation used in MotionCtrl ([127]), prove to be ineffective in this context. Although Plücker coordinates used by ([5, 35, 130]) may mitigate the problem slightly but our study reveals it is the camera embedding dimension that plays a more significant role. Improvement can be achieved with appropriate method to project the camera parameters into higher dimension.

Despite these improvements, DiT still experiences significant deterioration in camera motion, even with an optimal combination of camera representation and conditioning methods. Implementing two state-of-the-art U-Net methods ([35, 127]) in DiT resulted in videos with static or limited camera motion and less accurate camera orientations. To address this, we propose a novel Camera Motion Guidance (CMG) method based on the widely used classifier-free guidance technique [42]. CMG improves camera pose accuracy and motion over 400% compared to baseline DiT models. Being architecture-agnostic, it applies generically to both space-time and spatio-temporal architecture, in either DiT or U-Net models to improve camera control in video generation.

On the other hand, existing methods for camera control rely on dense camera input, requiring a camera pose for every frame, which is tedious, especially for long videos. To simplify this process, we propose a novel data augmentation pipeline that introduces sparse camera control, where only the camera pose for the final frame or a few key interval frames is required. Our experiments demonstrate that sparse camera control shown promising results, simplifying the input process while maintaining high-quality results and precise camera motion.

We summarise our main contributions as below:

- We introduce novel Camera Motion Guidance, a classifier-free guidance method, improving camera motion by over 400% in video diffusion transformers.
- We identify the root causes of camera control degradation in DiT and successfully developed the first camera control model for space-time video diffusion transformers.
- We develop a novel data augmentation pipeline that enables sparse camera control, which simplifies the required input control signal. To our knowledge, this is a unique feature not demonstrated in the existing work.

6.2 Related Works

Video Diffusion Models. Diffusion models [40, 84, 95, 112] have achieved remarkable success in image generation, leading to advancements in video diffusion models that build upon this foundation. The video diffusion models [8, 43] first process individual frames by applying 2D latent encodings to each image separately, and then fuse the temporal dimension to generate videos, this is known as space-time encoding (2D+1D). More recent developments [59, 68, 73, 148] have introduced spatio-temporal encoding, which encodes multiple frames simultaneously (3D), creating more compact latent code for long video generation.

Camera Pose Control has existed since early 2D human image generation methods. By modifying the size and vertical rotation of skeleton images, methods such as [18, 53, 72, 91] could influence camera pose, though only indirectly in limited ways. As neural network architectures have evolved from convolutional networks [60] to transformers [120], parameterised poses have been explored as substitutes for pose images [16]. In a study, [17] explicitly mapped 3D camera translation parameters along with 3D SMPL body pose parameters [69], allowing for simultaneous control of both body pose and camera pose.

As video generation emerged as active research, attention has been focused on controlling camera motion for video. AnimateDiff [33] trains module on specific motion and hence requiring new module for a different motion, hindering usability. Hence, **parameteric camera control** has recently become a focal point in video generation to improve its ease-of-use. Direct-a-video [134] uses only translation movement limiting the camera motion to only panning and zooming. Instead, MotionCtrl [127] introduced a rotation-and-translation matrix derived from camera extrinsic parameters, allowing more complex motion. CameraCtrl [35], on the other hand, uses Plücker coordinates as camera pose representation, enabling geometric interpolation for each pixel. We evaluate both state-of-the-art parameterisation methods to assess their effectiveness in DiT-based models. In these methods, camera poses are applied to individual image latents on a frame-by-frame basis, which makes them unsuitable for direct application to spatio-temporal models such as concurrent works [5, 146] where multiple frames are jointly encoded into a single latent representation. To the best of our knowledge, we are the first to develop camera control methods specifically for space-time DiT, enabling video models to leverage the extensive availability of pretrained 2D diffusion models.

Diffuser Guidance. [22] demonstrated that applying guidance in diffusion models significantly enhances image quality. During inference, in each denoising step, the model denoises the latent twice—once without conditioning and once with it. The difference between the conditioned and unconditioned latents defines a direction that can be extrapolated by multiplying it with a scalar guidance scale. In particular, classifier-free guidance ([42]) has become ubiquitous in modern text prompting diffusion models in which an empty string or negative text prompt is used as unconditional reference to provide latent extrapolation to improve image or video quality. Despite advancements in camera control for video generation [5, 35, 127, 130], diffuser guidance has not been explored for camera motion improvement. In this chapter, we introduce camera motion guidance to enhance the accuracy and quality of camera motion in video generation models.

Sparse Camera Control. Existing methods rely on dense camera poses to achieve effective control.

SparseCtrl [32] explores applying sparse image-based structural control but does not incorporate camera control, leaving a gap in addressing sparse camera pose scenarios for video generation tasks.

6.3 Our Approach

6.3.1 Preliminary

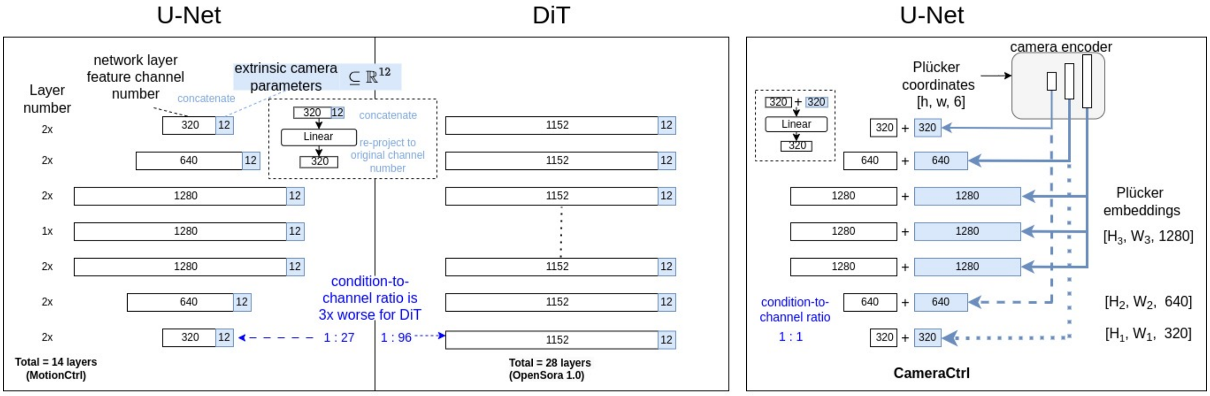
Camera Representation. Extrinsic camera parameters describe the camera's position and orientation in 3D space represented by a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{T} \in \mathbb{R}^3$ which form the rotation-and-translation (RT) matrix $[\mathbf{R}|\mathbf{T}] \in \mathbb{R}^{3\times4}$. Intrinsic parameters, encapsulated in the camera matrix $\mathbf{K}$, define the camera's internal characteristics, including focal length, principal point and pixel size. These are used to map a 2D pixel location in the image to a 3D direction vector in camera's coordinate system. For each pixel (x, y) in image coordinate space, its Plücker coordinate is calculated as $(\mathbf{O} \times \mathbf{d}_{x,y}, \mathbf{d}_{x,y)}$ where $\mathbf{O} \in \mathbb{R}^3$ is the camera center in world coordinates derived from $-\mathbf{R}^\top\mathbf{T}$; and the direction vector $\mathbf{d} \in \mathbb{R}^3$ is obtained by:

$$\mathbf{d}_{x,y} = \mathbf{R}\mathbf{K}^{-1}[x, y, 1]^\top \tag{6.1}$$

This formulation represents the direction and location of 3D lines, allowing for efficient geometric operations like interpolation and transformation, which are useful for camera trajectory and ray-based rendering. Compared to a flattened RT $\in \mathbb{R}^{12}$ in a video frame, Plücker coordinates has higher dimension $\in \mathbb{R}^{h,w,6}$ where h and w are height and width of video resolution.

Camera Conditioning for Video Generation. We examine the state-of-the-art camera conditioning in U-Net models to understand how their network topology differences from DiT models affect the effectiveness of camera conditioning. In U-Net architecture, the spatial resolution decreases as the features traverses down the network, while the channel C increases from e.g. 320 to a maximum of e.g. 1280 before stepping down again. MotionCtrl([127]) employs a RT matrix to as camera pose representation for the entire video frame. To incorporate this into the U-Net, the flattened RT is repeated for every latent pixel and concatenated with the U-Net's features along channel dimension, forming a new dimension of C+12 where C is the U-Net embedding's channel number. However, increased spatial resolution of U-Net's embedding is accompanied with repetitive RT in each spatial position and does not carry more camera information. Thus we can ignore the spatial dimension when considering camera conditioning influence and consider only the channel dimension. As illustrated in Figure 6.2(a), U-Net has the smallest embedding channel at its top and bottom layers, resulting in the highest ratio of camera parameter dimension to embedding dimension, we refer to as *condition-to-channel ratio*. For simplicity, we display only the channel dimension of the full tensor shape $[B, N, H, W, C]$ (batch size, number of frames, image height, image width, channel) and omit the other dimensions in the figure.

In contrast, transformers maintain a consistent channel number e.g. 1024 across all the layers. When implementing MotionCtrl's method to OpenSora ([148]) DiT, the increase of minimum channel number from 320 to 1152 lead to significant drop (3×) of the condition-to-channel ratio of 12:C from 1:27 to 1:96. We hypothesize camera conditioning strength is proportionate to this ratio, which leads to a considerably weakened control strength in DiT.


U-Net
DiT
U-Net
network layer feature channel number
extrinsic camera parameters $\subseteq \mathbb{R}^{12}$
Layer number
concatenate
re-project to original channel number
Linear
2x
1x
320
640
1280
1152
12
condition-to-channel ratio is 3x worse for DiT
1 : 27
1 : 96
Total = 14 layers (MotionCtrl)
Total = 28 layers (OpenSora 1.0)
camera encoder
Plücker coordinates [h, w, 6]
Plücker embeddings
[H$_3$, W$_3$, 1280]
[H$_2$, W$_2$, 640]
[H$_1$, W$_1$, 320]
condition-to-channel ratio
1 : 1
CameraCtrl

(a) MotionCtrl Method (RT, concatenation)

(b) CameraCtrl Method (Plücker, add)

Figure 6.2: We illustrate two prominent camera control U-Net model architecture (a) concatenation of rotational-and-translation matrix, and (b) addition of Plücker embedding. For clarity, only channel dimension is shown, omitting batch, frame, height and width of features. Comparing with DiT, U-Net has a worse condition-to-channel ratio leading to weakened control.

Our experiment compares another conditioning scheme from CameraCtrl ([35]) to confirm our hypothesis. As Plücker coordinates has dimension of $[h, l, 6]$ per camera pose, a camera encoder is necessary to produce multi-scale camera embedding that matches the dimensions of the DiT's features in each layer l, represented as $[H_l, H_l, C_l]$ for element-wise addition as shown in Figure 6.2b. This difference is even more pronounced when considering the spatial dimension, as each Plücker embedding is unique for each spatial location, thereby offering significantly stronger camera conditioning. At a high level, the main differentiators between the methods are the camera representations and the approach to fusing camera embeddings: MotionCtrl concatenates RT, while CameraCtrl adds Plücker embedding to the U-Net features.

Both methods share several common practices. First, each fused embedding is projected back to the original embedding dimension via a linear layer at every U-Net layer. Additionally, camera conditioning is applied before the temporal transformers. Only the temporal transformers and the newly added camera control components are updated during training, while all other parameters remain frozen. These practices were similarly adopted in our experiments. Another method, CamCo ([130]), employs a conditioning scheme similar to CameraCtrl, with the key difference being the use of 1×1 convolution layers instead of linear layers for projection. However, due to the lack of open-source implementation, we have not experimented with it in our study.

6.3.2 DiT Baseline Model Implementation

We adopts an open source DiT text-to-video (T2V) model OpenSora ([148]) as our base video generation model. They have newer versions that employ a spatio-temporal 3D encoder, enabling higher resolution and better image quality. However, we use OpenSora 1.0 that uses space-time architecture, allowing for direct comparison with the U-Net models. Then OpenSora 1.0 model consists of a cascade of 28 Spatial-Temporal DiT (ST-DiT) blocks inspired by [73], each containing a spatial transformer followed by a temporal transformer.

We implemented two methods as our DiT baselines: DiT-MotionCtrl and DiT-CameraCtrl, named after their respective U-Net-based counterparts. The methods were faithfully implemented for like-for-like comparison, with adjustments to the channel dimension of linear layers and the camera encoder to match the embedding dimension of DiT. Additionally, given DiT's uniform channel dimension, we have only one block in the camera encoder, in contrast to three blocks in U-Net's producing camera embeddings at three different resolutions.

6.3.3 Data Processing and Camera Augmentation

Given that our base video model is limited to 16 frames, translating to about 0.5 seconds of footage at 30 frames-per-second, taking consecutive frames would result in minimal camera motion. Therefore, we extract video frames from the dataset in strides of 4 to 8 frames, sampled uniformly, ensuring we capture sufficient temporal information while preserving meaningful motion dynamics. We randomly sample 16 frames from a training video sample, and this can produce arbitrary large starting translation vector. Therefore, we represent all RT matrices relative to the first frame by setting the translation vector $\mathbf{T}_1 = \mathbf{0}_3$ and rotation matrix $\mathbf{R}_1 = \mathbf{I}_{3\times 3}$ and multiplying $[\mathbf{R}_1|\mathbf{T}_1]^{-1}$ to the rest of the RT matrices.

To ensure smooth integration of our novel camera control method, we now establish a robust camera augmentation pipeline. Inspired by [111], we randomly drop out camera poses in a video by setting them to zeros to prevent overfitting and allow for sparse camera control, we call this *zero camera*. In training, we randomly generate *static video* by repeating the first frame and its corresponding camera pose across all subsequent frames. In other words, all camera poses are filled with value of $[\mathbf{R}_1|\mathbf{T}_1]$ or its corresponding Plücker coordinate, a condition we denote as *null camera*, $\emptyset_C$. In addition to the new proposed augmentation, we also adopted standard video augmentation of center image cropping and video temporal reversal which is critical as it balances the distribution of two opposite camera motions.

6.3.4 Camera Motion Guidance (CMG)

We now introduce our novel method - Camera Motion Guidance- based on classifier-free guidance. Equation 6.2 shows the generic classifier-free guidance equation for text prompt ([42]).

$$\hat{e_\theta}(z_t, C_T) = e_\theta(z_t, \emptyset_T) + s_T\{e_\theta(z_t, C_T) - e_\theta(z_t, \emptyset_T)\} \tag{6.2}$$

where e_θ is the denoising model (U-Net or DiT), z_t is the noisy latent at time step t, C_T is the text condition, $\emptyset_T$ is null text condition and s_T is text prompt guidance scale. In essence, the second term in the equation finds the text embedding direction in the latent space and extrapolates it with a scalar s_T, using the unconditioned first term as a reference point.

In existing camera-controlling literature [35, 127], classifier-free guidance is applied solely to the text prompt, with the camera condition C_C present in all guidance terms. This approach is analogous to Equation 6.3, effectively cancelling out the camera condition's influence in the second term.

$$\hat{e_\theta}(z_t, C_T, C_C) = e_\theta(z_t, \emptyset_T, C_C) + s_T\{e_\theta(z_t, C_T, C_C) - e_\theta(z_t, \emptyset_T, C_C)\} \tag{6.3}$$

To address this, we propose a new camera motion guidance term, disentangling it from the original text guidance term, as outlined in Equation 6.4. Following the principles of CFG, the zero camera would typically serve as an unconditional camera control signal. However, in our approach, we repurpose the zero camera as a dropout mechanism for camera poses. Instead, we introduce null camera $\emptyset_C$, representing a static video as a reference. The camera motion guidance scale, s_C, is then applied to guide the camera motion independently. This design ensures that camera motion is guided effectively while maintaining flexibility in handling sparse camera poses.

$$\begin{aligned}\hat{e_\theta}(z_t, C_T, C_C) = e_\theta(z_t, \emptyset_T, \emptyset_C) + s_T\{e_\theta(z_t, C_T, \emptyset_C) - e_\theta(z_t, \emptyset_T, \emptyset_C)\} \\ + s_C\{e_\theta(z_t, C_T, C_C) - e_\theta(z_t, C_T, \emptyset_C)\}\end{aligned} \tag{6.4}$$

With simple changes in data augmentation, our method CMG can enhance any video generative model employing classifier-free guidance, offering improved camera control across a variety of architectures.

6.4 Experiments

6.4.1 Implementation Details

We train our models using the RealEstate10k dataset [149], which features indoor and outdoor real estate videos with corresponding camera poses. The models are trained at a resolution of 256×256 and 16 frames per video sample, matching the settings of the U-Net models for comparisons. Since the original dataset lacks captions, we use the text prompts provided by [35]. The training was conducted on GPUs with 40GB memory, using a batch size of 3 per GPU, and models were trained for 8 epochs with a fixed learning rate of 1×10^{-5}.

We evaluate two baseline DiT models: DiT-MotionCtrl and DiT-CameraCtrl, which use RT matrices and Plücker coordinates as their respective camera representations. Additionally, to enable CMG, we train the same models with our augmentation pipeline which saw 5% null camera data augmentation. Both the baseline and CMG versions also applied a 5% camera dropout, which is randomly set between 70% and 100% of camera frames in a video to zero.

During inference, the two baseline models use standard guidance (Eq. 6.3), while CMG models apply Eq. 6.4. A text guidance scale s_T of 4.0 is used consistently across all the DiT models. We evaluate a range of CMG scale s_C between 4.0 and 7.0, eventually selecting 5.0 for comparison with the baselines. Apart from this, identical configurations and random seeds are applied to all DiT models to ensure fair comparisons. We use U-Net models' default configurations for inferences. To investigate the effect of camera representation further, we also replace the Plücker coordinates in DiT-CameraCtrl with RT matrices and re-train a new model.

6.4.2 Metrics

We aim to measure two aspects of camera motion: first, the accuracy with which the camera motion adheres to the specified camera conditioning and second, the extent of motion present in the generated videos. For the camera motion accuracy, we adopt approach from [35] to utilise COLMAP [102] in extracting rotation matrices $\mathbf{R}_{gen} \in \mathbb{R}^{N\times3\times3}$ and translation vectors $\mathbf{T}_{gen} \in \mathbb{R}^{N\times3}$ of generated videos where N is frame length. The rotation error $\mathbf{R}_{err}$ and translation error $\mathbf{T}_{err}$ are calculated by comparing with ground truth $\mathbf{R}_{gt}$ and $\mathbf{T}_{gt}$ respectively using Eq. 6.5 and 6.6.

$$\mathbf{R}_{err} = \sum_{n=2}^{N} \cos^{-1}\left(\frac{tr(\mathbf{R}^{n}{}_{gen}\mathbf{R}_{gt}^{n\top}) - 1}{2}\right) \tag{6.5}$$

$$\mathbf{T}_{err} = \sum_{n=2}^{N} \|\mathbf{T}_{gt}^{n} - \mathbf{T}_{gen}^{n}\|_2 \tag{6.6}$$

where n is n-th video frame and tr is the trace of a matrix. We exclude the calculation of error for the first frame, as it is always zero by definition due to the camera poses preprocessing. We report the rotation error in radian. The translation range in generated videos can vary, but more importantly, COLMAP estimation can yield a wide translation range. Therefore, we normalise both translation vectors $\mathbf{T}_{gt}$ and $\mathbf{T}_{gen}$ to have a unit maximum distance during inference for metric evaluation.

To quantitatively assess the level of motion in the generated videos, it is essential to identify an appropriate measurement method. After considering various options, we ultimately select [114] to measure the optical flow between two frames, which gives a flow field represented as two arrays u and v corresponding to each pixel's horizontal and vertical components of motion. The motion magnitude M is then calculated for all adjacent frames using Eq. 6.7, and the results are averaged over the entire video.

$$M = \frac{1}{K}\sum_{k=1}^{K}\sqrt{u_k^2 + v_k^2} \tag{6.7}$$

where k is k-th pixel in a video frame and K is total pixel counts in a frame. We also use FID ([37]) to measure image quality in the ablation study, ensuring that our method maintains high video quality.

6.4.3 Results

Motion Degradation with DiT Implementation. Porting the MotionCtrl method into DiT leads to a total loss of controllability, as also observed by [5]. This is evident in sharp rise in rotation error as shown in Table 6.1 (Model 1a→1b). Most importantly, the motion magnitude collapses 80% from 7.780 to 1.485. The lack of motion is difficult to perceive from still images, therefore we included "Supplementary Video 1 - Method Comparison" [1] to demonstrate the stark contrast in motion. Although applying CMG to DiT-MotionCtrl (Model 1c) brings slight improvement(compared to Model 1b), the high errors and the lack of motion persist, indicating the corresponding U-Net method is not effective for DiT.

Camera Motion Guidance Restores Controllability and Significantly Boosting Motion. While the baseline DiT-CameraCtrl's rotation error remained at a similar level, meaning it has better controllability, it still suffered from severe losses in motion and translation accuracy (Model 2a→2b) as illustrated in Figure 6.3 and "Supplementary Video 2 - Ablation" [2]. Applying CMG significantly boosted both metrics, with a notable 412% increase in motion magnitude, from 1.564 to 6.450 (Model 2b→2c). Our model (Model 2c) significantly outperformed both DiT baseline models (Model 1b and 2b) and also surpassed both U-Net models in translation error, which is a more critical metric for our study than rotation error. It is worth noting that, motion as measured from optical flow is sensitive to

[1] https://github.com/soon-yau/CameraMotionGuidance/tree/web/supplementary/1_Method_Comparison.htm

[2] https://github.com/soon-yau/CameraMotionGuidance/tree/web/supplementary/2_Ablation.htm

Model	Camera	RotErr ↓	TransErr ↓	Motion ↑
(1a) MotionCtrl (U-Net)	RT	0.168	0.640	7.786
(2a) CameraCtrl (U-Net)	Plücker	0.176	0.754	9.686
(1b) DiT-MotionCtrl (baseline)	RT	0.224	0.716	1.485
(1c) DiT-MotionCtrl w CMG (Ours)	RT	0.208	0.702	1.806
(2b) DiT-CameraCtrl (baseline)	Plücker	0.186	0.687	1.564
(2c) DiT-CameraCtrl w CMG (Ours)	Plücker	**0.176**	**0.577**	**6.450**

Table 6.1: Quantitative results showing the performance degradation of camera conditioning implemented for DiT architecture. Our method of applying CMG to DiT-CameraCtrl significantly improves the metrics, outperforming all DiT baselines.

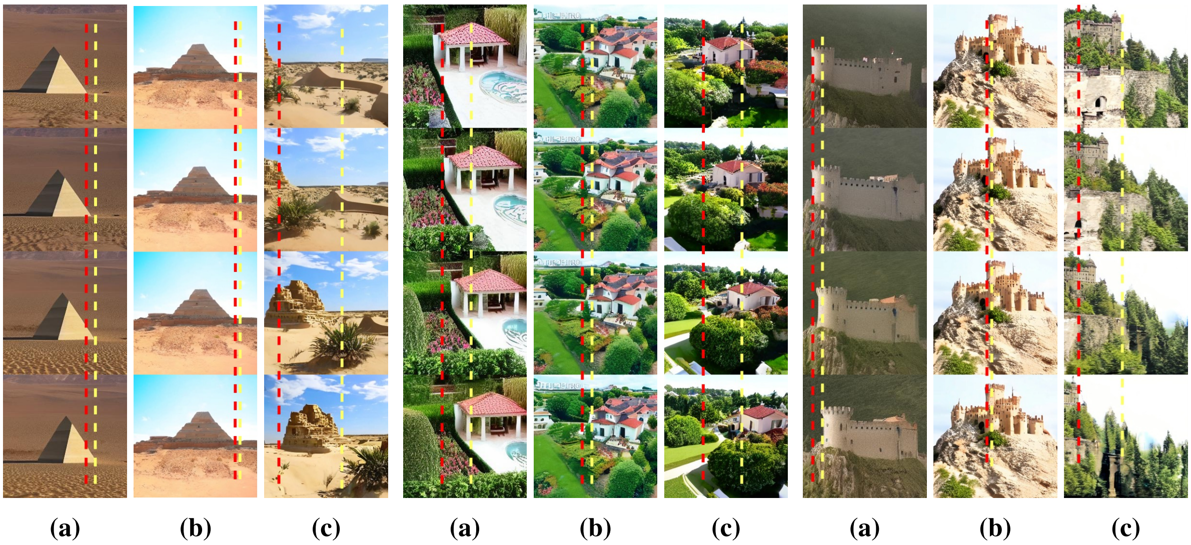


Figure 6.3: (a) CameraCtrl (b) DiT-CameraCtrl (c) DiT-CameraCtrl w CMG (Ours). Direct DiT implementation of CameraCtrl results in severe loss of camera motion in specified pan left/right camera condition. However, applying our method CMG restore the controllability and boosting the camera motion.

pixel quality and video content. Therefore, the motion magnitude is not directly comparable with U-Net models which are based on different pre-trained base models and training recipes.

Comparing Plücker Coordinates and Extrinsic Parameters. Previously in Table 6.1, we have shown that DiT-CameraCtrl (Model 2b) is more effective than DiT-MotionCtrl (Model 1b), and the performance gap becomes even more significant with our CMG (Model 1c→ $2c$). However, aside from their conditioning methods, the two methods also use different camera data representation. To make a fair comparison, we repeated the DiT-CameraCtrl experiment by replacing the Plücker coordinates with RT matrices. This allowed us to isolate and compare the methods with only one differentiating factor at a time.

From Table 6.2, we can draw three key insights: first, RT is not inherently inferior, as DiT-CameraCtrl achieves better overall results with RT (Model 2d) compared to Plücker coordinate (Model 2b). Secondly, conditioning method plays a crucial role. DiT-CameraCtrl consistently outperforms DiT-MotionCtrl for both camera representation (Model 2b,d vs 1b), confirming our hypothesis that a better condition-to-channel ratio strengthens camera conditioning. Lastly, our CMG method consistently boosted DiT-CameraCtrl's performance in both Plücker coordinate (Model 2b→2c) and RT (Model 2d→ 2e), and

Model	Camera	RotErr ↓	TransErr ↓	Motion ↑
(1b) DiT-MotionCtrl	RT	0.224	0.716	1.485
(2b) DiT-CameraCtrl	Plücker	0.186	**0.687**	1.564
(2d) DiT-CameraCtrl	RT	**0.177**	0.748	**2.101**
(2c) DiT-CameraCtrl w CMG	Plücker	**0.176**	**0.577**	**6.450**
(2e) DiT-CameraCtrl w CMG	RT	0.177	0.666	5.721

Table 6.2: Comparing individual effect of model and camera representation. Results for Model 1b, 2b, 2c are included from Table 6.1 for ease of comparison.

also improve DiT-MotionCtrl (Table 6.1:1b→1c). This demonstrates the robustness and effectiveness of CMG in improving camera control across different configurations.

6.4.4 Ablations

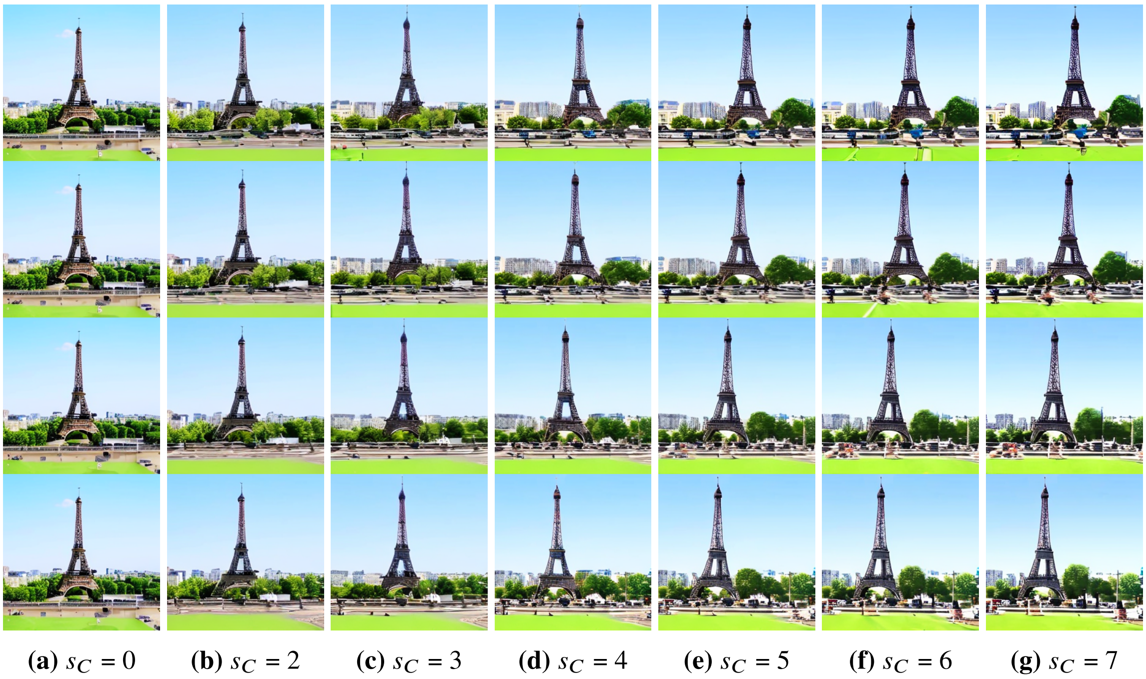


Figure 6.4: Each column is showing a video generated with a different CMG scale s_C. As the scaling increases, the specified camera motion (pan right) is also more pronounced, evident from greater translation in the Eiffel Tower's position. The visually similar videos also demonstrate effective disentanglement of our camera motion guidance from the text guidance term.

We conduct an extensive study varying the CMG scale across a range of values to prove its effectiveness in inducing and controlling camera motion. As illustrated in Figure 6.4, increasing the CMG scale results in greater camera motion. The preservation of video content highlights the strong disentanglement of our method, allowing independent camera motion control, separated from the text guidance in classifier-free guidance (Eq. 6.4) that describes the video scene.

As DiT-MotionCtrl has proven ineffective, we focus our discussion on DiT-CameraCtrl for the quantitative result shown in Table 6.3. Determining the optimal CMG scale is not straightforward, as no single value consistently delivers the best results across all metrics. Additionally, each metric has its own limitations, making the selection of an ideal CMG scale more nuanced. While higher motion

	Rot error ↓	Transl error ↓	Motion ↑	FID ↓ (vs CameraCtrl)
DiT-CameraCtrl	0.186	0.687	1.564	91.3
DiT-CameraCtrl w CMG= 4	0.172	0.595	5.721	**69.5**
DiT-CameraCtrl w CMG= 5	0.176	**0.577**	6.450	70.5
DiT-CameraCtrl w CMG= 6	0.180	0.597	7.061	71.3
DiT-CameraCtrl w CMG= 7	**0.169**	0.600	**7.631**	73.8

Table 6.3: Ablation of different CMG scales.

magnitude increases movement, it can also result in blurrier videos, as reflected by the degradation in FID scores compared to those produced by U-Net model using the same text prompt. Among the error measurements, translation error plays a more critical role in typical camera motion scenarios. Therefore, we selected CMG scale of 5.0 which minimises translation error for optimal performance, and used it for main comparison in Table 6.1.

6.4.5 Sparse Camera Control

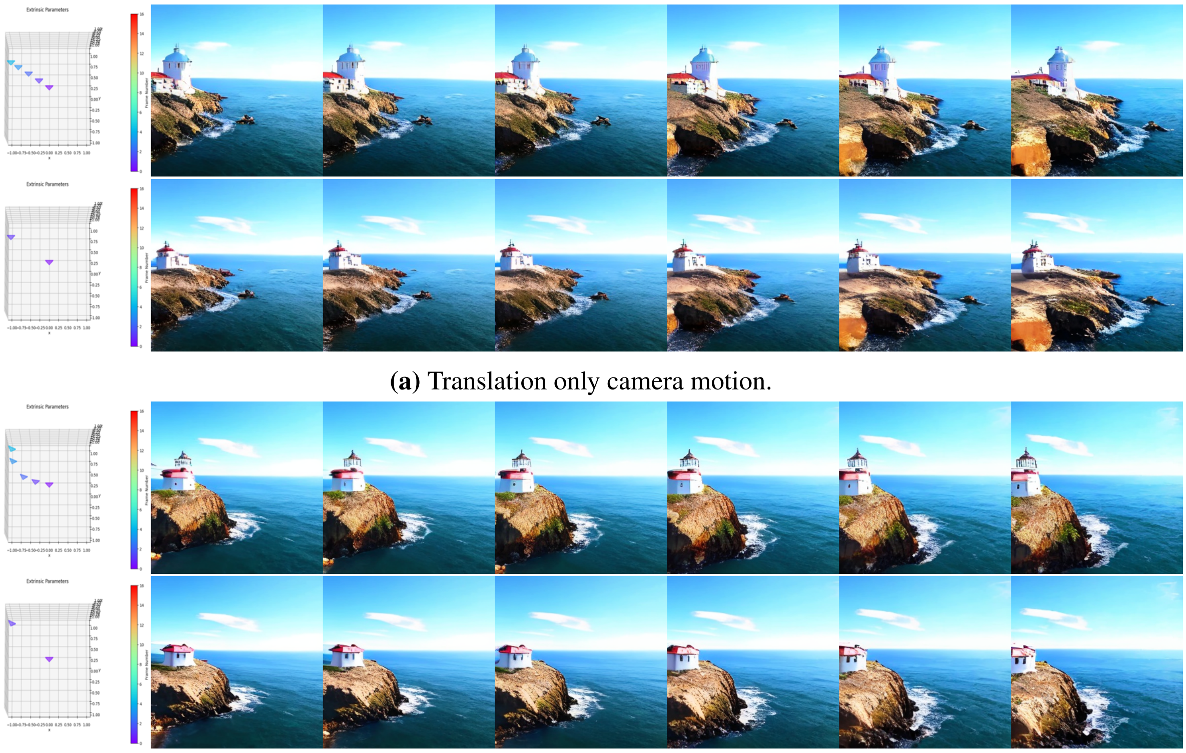
Since we drop certain interval frames by setting the camera poses to zeros during training, our method allows users to provide camera control for only a sparse set of frames at test time, which, to our knowledge, is not supported by existing methods. Translation motion in a single dimension, such as zooming, can be easily interpolated and does not offer significant value in testing sparse control. Therefore, we excluded simple translation motion from evaluation. Table 6.4 presents the rotation and translation errors at different sparsity ratios, which we define as the ratio of dropped camera poses to $N - 1$ frames, excluding the first frame. While errors do increase with higher sparsity ratios, the rate of error increase remains relatively modest compared to the level of sparsity, even up to 87% sparsity, where only the camera poses in the first, middle, and last frames are provided.

# camera poses dropped	Sparsity Ratio	Rot error ↓	Transl error ↓
0	0%	**0.176**	**0.611**
7	47%	0.181	0.627
11	73%	0.203	0.650
13	87%	0.218	0.755
14	93%	0.291	0.768

Table 6.4: Increasing camera control sparsity results only in modest drop in controllability as measure dby rotation and translation error.

Figure 6.5 shows videos generated using sparse camera poses as specified in the leftmost column where only 4 frames (73% sparsity) and 1 frame (93% sparsity) are used respectively. When only the last frame was used, the specified camera pose Figure 6.5a and Figure 6.5b end in similar position, differing only in the camera rotation. In Figure 6.5a, the generated camera motion accurately follows the translation-only trajectory, maintaining a straight-facing camera angle as expected. On the other hand, the rotated ending camera pose in Figure 6.5b result in a smooth rotating motion alongside translation, similar to the video above generated using denser camera poses. The videos, which can be viewed in "Supplementary Video 3: Sparse Control" [3] highlights our model's ability to interpolate the camera

[3] https://github.com/soon-yau/CameraMotionGuidance/tree/web/supplementary/3_SparseControl.htm



(a) Translation only camera motion.

(b) Simultaneous translation and rotation camera motion.

Figure 6.5: Videos following specified camera trajectories with sparse camera control as shown in the left.

poses to fill in the gaps and maintaining smooth, coherent camera motion. Our sparse camera data augmentation technique is also effective with standard DiT methods without CMG. While this model demonstrates weaker camera controllability and motion without CMG, it still successfully interpolates camera poses, proving its robustness in sparse control scenarios.

While camera pose interpolation can be achieved by interpolating camera parameters, our sparse camera control method offers a practical advantage in data collection for training. In real-world scenarios, pose estimation methods such as COLMAP are often unreliable, especially in challenging environments with textureless surfaces, motion blur, or occlusions. These limitations can lead to incomplete pose estimates, where some frames lack valid camera parameters. As a result, valuable video samples may need to be excluded from the dataset, reducing the diversity and quantity of training data. By leveraging sparse camera control, we can mitigate this issue by ensuring that only a few reliable keyframe poses are needed, allowing for interpolation to reconstruct missing poses. This not only increases the usability of imperfect datasets but also enhances the robustness of training by incorporating a broader range of video samples.

6.5 Limitations

Although object motion is excluded from our study, it is not negatively impacted by CMG. In "Supplementary Video 4 - Object Motion", we showcase object motion from natural landscapes and 3D character animation alongside camera motion controlled using our method. However, the videos generated by our models may show limitations in image quality and content richness compared to models pre-trained on larger datasets. The OpenSora 1.0 we use was pre-trained on 400K video clips—a much smaller dataset than the 10M videos used by MotionCtrl. This constraint may also have led to occasional deformations for objects such as the Eiffel Tower, which are not present in the RealEstate10k dataset we trained on. Since our CMG method effectively disentangles camera motion from the text prompt, we believe more visually appealing videos could be generated with a higher-quality base video model.

6.6 Conclusions

This chapter thoroughly examined the impact of various camera representations and conditioning methods on camera control for video generative diffusion transformers. Our extensive experiments confirmed that high-dimensional camera embedding is critical for effective camera control, supporting our hypothesis. We also found that camera representation alone is not the key factor for successful control; instead, it must be paired with effective conditioning methods and guidance techniques to achieve optimal results. We successfully demonstrated the first camera control model for space-time DiT by combining the CameraCtrl architecture, Plücker coordinates for camera representation, and our novel camera motion guidance (CMG). We have proven that CMG is highly effective in inducing motion and enhancing camera control. Due to limited resources and code availability, we could not experiment with CMG on a broader range of video models. However, we believe it would be equally effective for U-Net and other DiT models with spatio-temporal architectures. Additionally, we introduced novel camera data augmentation techniques that enable sparse camera control. These simple yet effective methods are generic, making them applicable and beneficial for a broader range of video model architectures.

In future work, we aim to test our CMG method on a wider range of models, including U-Nets and other spatio-temporal DiT. Additionally, we plan to enhance our approach to sparse camera control, ensuring that it can achieve even greater accuracy in interpolating camera poses.

Chapter 7

Conclusion

7.1 Ethical Implications

The ethical implications of the research presented in this thesis, particularly the generation of modified images of humans, warrant careful consideration. While the advancements in controllable image and video generation offer tremendous potential for creativity, entertainment, and various applications, they also raise concerns about privacy, consent, and the potential for misuse. The ability to modify human likenesses could lead to the creation of deepfakes or other misleading content that can be harmful, especially if used maliciously to deceive or manipulate individuals. Furthermore, there are challenges related to the representation of diverse identities and the potential for reinforcing harmful stereotypes if the generated content is not carefully managed. To mitigate these risks, incorporating watermarking techniques into the generated content could provide a means of tracking and verifying its authenticity, ensuring transparency and accountability. It is crucial that the technologies developed in this research be used responsibly, with an emphasis on informed consent, and robust safeguards to prevent unethical use. Future work should address these concerns by incorporating such mechanisms and promoting the ethical use of generated content.

7.2 Summary of Contributions

This thesis presents novel approaches to controllable image and video generation, with a particular emphasis on pose conditioning and the integration of multimodal inputs such as text and images. Building on three core conditioning techniques — direct feature fusion, attention mechanisms, and diffuser guidance — we introduce innovative methods and applications that enhance the controllability of generative models. These advancements tackle emerging challenges associated with the evolving architectures of generative models, ensuring a more precise alignment between user intent and model output. Overall, successful integration of more input conditions led to more fine-grained control, as evidenced in versatility of various human image generation tasks. The following sections provide a summary of the key contributions, highlighting the techniques and methodologies developed to advance controllable multimodal conditioning in image and video generation.

7.2.1 Parametric Pose Tokens

As generative model trends shifted from GANs to tokenised transformer-based systems, we observed that representing pose using skeleton images became less effective. Unlike GANs, which process continuous image inputs, transformer-based models operate on discrete tokens, making direct image-based pose conditioning suboptimal. To address this, we devised pose tokens, a structured representation that seamlessly integrates with the tokenized architecture, enabling more precise and efficient control over human pose in image and video generation. We successfully experimented with pose tokens derived from both 2D and 3D body pose representations, integrating them into autoregressive transformers and diffusion models. In our experiment with autoregressive transformers, we conditioned pose by concatenating input text tokens with pose tokens for self-attention, allowing the model to effectively capture human motion cues. However, applying the same method to latent diffusion models proved insufficient in ensuring spatial accuracy, as the model struggled to consistently align the generated pose with the intended structure. To address this issue, we introduced a silhouette mask at the input of the U-Net denoising network using direct feature fusion, enforcing spatial alignment and improving the model's ability to generate precise and coherent human figures.

Our parametric method enhances the flexibility and granularity of pose control, enabling the simultaneous interpolation of both body and camera poses—an unprecedented advancement in control from direct pose parameters. This breakthrough in 2D image generation not only improves flexibility and controllability but also lays the foundation for our later work in sparse camera control in video generation. Furthermore, parametric pose control eliminates the computational expensive step of skeleton image encoding, result in reduced computational requirements and improved speed.

7.2.2 Solving Mode Conflict

The use of adapters in pre-trained diffusion models has gained significant traction, particularly for incorporating new control mechanisms. However, conflicts often arise between the adapter branches and the base model. Despite the widespread challenges, especially in systems employing multiple adapters, this phenomenon has largely been overlooked by the research community. In our study, we systematically investigate this issue and introduce a new term — *mode conflict* —to describe the control imbalance when conflicting training data between the adapters and the base model lead to a control mode dominating over the others. To address this issue in multi-conditioning of pose, text and image, we proposed ViscoNet, a unified adapter approach, eliminating separate pose control adapter e.g. ControlNet and image adapter e.g. IP-Adapter. In ViscoNet, we employs cross-attention to integrate visual and pose conditioning before control signals are generated, allowing for smooth multimodal conditioning. Together with other innovations e.g. control feature masking and multiscale control, our method offers flexibility in achieving harmonious balance between different control mechanisms.

7.2.3 New Evaluation Methods

Throughout our research journey, we encountered new challenges that had not been addressed in the existing literature. As there were no established evaluation metrics for these issues, we developed new ones to effectively assess and quantify the problems. We introduced *PCE (People Count Error)*, a metric that effectively identifies the unique artifacts in generated human images. This method has been positively received and adopted by the research community [53, 65, 137] as a benchmark for evaluating the accuracy of human image generation. Additionally, we proposed a robust evaluation metric *Mode Conflict Rate (MCR)* to quantify the degree to which diverse conditioning inputs maintain their intended characteristics, providing a reliable measure of controllability and stability in multi-conditioning generative models.

7.2.4 Camera Motion Guidance (CMG)

Transformer-based diffusion models (DiTs) are rapidly becoming the preferred architecture for video generation due to their scalability and enhanced performance. However, porting existing camera control mechanisms from U-Net-based models into these frameworks resulted in a substantial loss of control. Through comprehensive experimentation, we identified an optimal conditioning method that relies on an effective camera representation to achieve dynamic and nuanced camera control. We found that a dedicated camera encoder that projects Plücker coordinates into high-dimensional embeddings is necessary in controlling DiT-based video models.

However, achieving practical camera pose control required our novel *Camera Motion Guidance*—a unique diffuser guidance method. We introduced a new concept of "null camera pose" and static video which serves as an anchor point for extrapolating camera motion. Our method outperform the baseline methods and boost camera motion by over 400% as measured using optical flow. Additionally, our data augmentation pipeline enables sparse camera conditioning by interpolating camera poses, drawing from insights gained in our previous work on human pose interpolation for 2D images. This pipeline simplifies user interaction, enabling comprehensive camera motion with minimal input. More importantly, the integration of sparse camera control during model training allows the inclusion of video samples lacking complete camera pose data due to imperfect pose reconstruction. The availability of more training video data can significantly improve model performance, allowing for more robust camera motion generation.

7.3 Future Works - Towards Multimodal Agent for Image Generation

Current multimodal models primarily operate in a one-way manner between modalities. For example, image-to-text (I2T) models can analyse images to generate descriptions and answer text-based queries, but they lack the ability to modify the input image based on textual instructions. Thereby, future advancements can be driven by insights from the rapid progress in LLMs, leveraging their capabilities in contextual understanding, reasoning, and adaptive learning to enhance multimodal generative control.

7.3.1 Zero-shot Retrieval-Augmented Generation (RAG)

Visual conditioning methods in Chapter 4 and 5 provides a robust way to guide human appearance and style generation. However, its fidelity is often restricted by the inherent biases and limitations within the model's pre-trained parameters. In other words, visual generation is only as accurate as the learned distributions of the original training data, which can limit specificity, especially when aiming to match particular styles or appearances not well-represented in the dataset. To overcome this limitation, future research could expand beyond fixed model knowledge, incorporating retrieval methods to leverage a larger set of external data. An example of memory retrieval is seen in [142], which retrieve clothing texture from a closed database by matching the closet embedding. Recently, retrieval-augmented generation (RAG) [61] enhances LLMs by allowing them to dynamically access and integrate external information, leading to more accurate, contextually relevant, and up-to-date responses beyond the model's static training data. Therefore, it is worth exploring using similar method for zero-shot visual reconstruction.

7.3.2 Memory Augmentation for Image Edit

While modern LLMs already exhibit long-term contextual understanding in text-based tasks, existing T2I diffusion models lack equivalent capabilities in maintaining a coherent history of edits and instructions. In Chapter 4, we introduced the use of conditioning images as memory in UPGPT[17] to facilitate interactive refinement with text, demonstrating early success in iterative image editing. A promising next step is to develop memory-augmented methods for diffusion models that can retain and utilise both textual and visual context over extended interactions. Additionally, an alternative direction worth exploring is the enhancement of multimodal autoregressive transformers, which have shown promise in maintaining structured long-term dependencies across different modalities.

7.3.3 Understanding Physical Properties

Future research should explore the ability to model physical world constraints beyond pixel-based representations in 2D images. For instance, it is crucial for models to understand that a close-up image of a toy basketball ball is physically smaller than a visually smaller basketball captured from a wide-angle shot, by leveraging spatial context and interactions with surrounding elements, such as a human figure. A promising avenue for addressing this challenge is the incorporation of 3D priors in combination with Reinforcement Learning with Human Feedback (RLHF). This approach could involve integrating 3D representations, such as depth maps and pose estimation, to estimate the physical dimensions of objects within an image. Furthermore, human feedback, such as the query “Which ball is bigger?” could be utilised to iteratively refine the model's understanding of spatial relationships and physical constraints, enhancing its ability to reason about object size and relative positioning in diverse scenes.

7.3.4 Text-Driven Image Structure Editing

Existing T2I diffusion models enable image editing via text prompts, but their capabilities are limited to basic modifications, such as replacing an object (e.g., changing an "orange" to an "apple") within a defined masked region. The mask is either explicitly provided [2, 3] or implicitly derived from attention maps [77]. However, these models lack the advanced reasoning, spatial awareness, and interpretative abilities of modern LLMs, making them incapable of executing more nuanced editing commands, such as “make the tree taller” or “place the tree in front of the house.” InstructPix2Pix [9] attempts to bridge this gap by leveraging automatically generated datasets containing image pairs with stylistic transformations and corresponding text instructions (e.g., “make it Picasso”). While effective for style transfer and texture alterations, this approach is fundamentally limited in its ability to modify object structure or scene composition. Extending this method to structural edits would require extensive manual annotation to create high-quality paired datasets, making it impractical at scale.

A promising research direction is to decompose image generation into two distinct components: structure and style. Structural modifications, such as repositioning or resizing objects, can be learned using physics-based simulation and rendering engines, where object transformations (e.g., location and scale adjustments) are paired with corresponding text instructions. This approach enables the generation of large-scale training datasets without the need for manual annotation. Once the structural edits are determined, a second-stage model can refine the image by synthesizing realistic textures and details based on real-world photographs. By leveraging this two-stage framework, we can significantly enhance the controllability and realism of text-driven image editing.

Bibliography

[1] Badour AlBahar, Jingwan Lu, Jimei Yang, Zhixin Shu, Eli Shechtman, and Jia-Bin Huang. Pose with style: Detail-preserving pose-guided image synthesis with conditional stylegan. *SIGGRAPH Asia*, 9 2021.

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transaction on Graphics*, 6 2022.

[3] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 11 2022.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[5] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. Vd3d: Taming large video diffusion transformers for 3d camera control. *arXiv preprint arXiv:2407.12781*, 7 2024.

[6] Shane Barratt and Rishi Sharma. A note on the inception score. *International Conference on Machine Learning (ICML)*, 2018.

[7] Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Jorma Laaksonen, Mubarak Shah, and Fahad Shahbaz Khan. Person image synthesis via denoising diffusion model. *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 11 2023.

[8] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[9] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[10] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *Conference on Neural Information Processing Systems (NeurIPS)*, 5 2020.

[11] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. *Proceeding of International Computer Vision Conference (ICCV)*, 4 2023.

[12] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis*

and Machine Intelligence, 2019.

[13] Hong Chen, Yipeng Zhang, Xin Wang, Xuguang Duan, Yuwei Zhou, and Wenwu Zhu. Disenbooth: Disentangled parameter-efficient tuning for subject-driven text-to-image generation. *arXiv preprint arXiv:2305.03374*, 2023.

[14] Wenhu Chen, Hexiang Hu, Yandong Li, Nataniel Rui, Xuhui Jia, Ming-Wei Chang, and William W Cohen. Subject-driven text-to-image generation via apprenticeship learning. *arXiv preprint arXiv:2304.00186*, 2023.

[15] Soon Yau Cheong, Armin Mustafa, Duygu Ceylan, Andrew Gilbert, and Chun-hao Paul Huang. Boosting camera motion control for video diffusion transformers. *Arxiv Preprint 2410.10802*, October 2024.

[16] Soon Yau Cheong, Armin Mustafa, and Andrew Gilbert. Kpe: Keypoint pose encoding for transformer-based image generation. In *British Machine Vision Conference (BMVC)*, 3 2022.

[17] Soon Yau Cheong, Armin Mustafa, and Andrew Gilbert. Upgpt: Universal diffusion model for person image generation, editing and pose transfer. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 4173–4182, 4 2023.

[18] Soon Yau Cheong, Armin Mustafa, and Andrew Gilbert. Visconet: Bridging and harmonizing visual and textual conditioning for controlnet. In *ECCV Workshops Proceedings*, 9 2024.

[19] Katherine Crowson. Clip guided diffusion. *https://github.com/crowsonkb*, 2020.

[20] Boris Dayma, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phuc Le Khac, Luke Melas, and Ritobrata Ghosh. Dall·e mini, 2021.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Arxiv Preprint 1810.04805*, 10 2018.

[22] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[23] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference for Learning Representations (ICLR)*, 2020.

[25] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *ICLR*, 2017.

[26] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[27] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. *ECCV*, 2022.

[28] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *ICLR*, 8 2022.

[29] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[30] Leon Gatys, Alexander Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[32] Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Sparsectrl: Adding sparse controls to text-to-video diffusion models. *arXiv preprint arXiv:2311.16933*, 2023.

[33] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *International Conference on Learning Representations*, 2024.

[34] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 2 2018.

[35] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 4 2024.

[36] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *Arvix Preprint 2208.01626*, 8 2022.

[37] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[38] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[39] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[40] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[41] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arxiv preprint:1912.12180v1*, 2019.

[42] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[43] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Neural Information Processing Systems (NeurIPS)*, 2022.

[44] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *ICML*, 2019.

[45] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv preprint arXiv:2311.17117*, 11 2023.

[46] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[47] HuggingFace. openai/clip-vit-large-patch14. *https://huggingface.co/openai/clip-vit-large-patch14*, 2011.

[48] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[49] Xuhui Jia, Yang Zhao, Kelvin CK Chan, Yandong Li, Han Zhang, Boqing Gong, Tingbo Hou, Huisheng Wang, and Yu-Chuan Su. Taming encoder for zero fine-tuning image customization with text-to-image diffusion models. *arXiv preprint arXiv:2304.02642*, 2023.

[50] Tao Jiang, Necati Cihan Camgoz, and Richard Bowden. Skeletor: Skeletal transformers for robust body-pose estimation. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021.

[51] Yuming Jiang, Shuai Yang, Haonan Qiu, Wayne Wu, Chen Change Loy, and Ziwei Liu. Text2human: Text-driven controllable human image generation. *SIGGRAPH*, 2022.

[52] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *European Conference on Computer Vision (ECCV)*, 3 2016.

[53] Xuan Ju, Ailing Zeng, Chenchen Zhao, Jianan Wang, Lei Zhang, and Qiang Xu. Humansd: A native skeleton-guided diffusion model for human image generation. In *International Conference on Computer Vision (ICCV)*, 4 2023.

[54] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 12 2018.

[55] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 12 2020.

[56] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations (ICLR)*, 2014.

[57] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[58] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Arxix Preprint Auto-Encoding Variational Bayes*, 12 2013.

[59] PKU-Yuan Lab and Tuzhan AI etc. Open-sora-plan, Apr. 2024.

[60] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[61] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*, 2021.

[62] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *International Conference on Machine Learning (ICML)*, 1 2023.

[63] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[64] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 2020.

[65] Sicheng Li, Keqiang Sun, Zhixin Lai, Xiaoshi Wu, Feng Qiu, Haoran Xie, Kazunori Miyata, and Hongsheng Li. Ecnet: Effective controllable text-to-image diffusion models, 2024.

[66] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. *European Conference on Computer Vision (ECCV)*, 5 2014.

[67] Xian Liu, Jian Ren, Aliaksandr Siarohin, Ivan Skorokhodov, Yanyu Li, Dahua Lin, Xihui Liu, Ziwei Liu, and Sergey Tulyakov. Hyperhuman: Hyper-realistic human generation with latent structural diffusion. *Arxiv preprint: 2310.08579*, 10 2023.

[68] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.

[69] Matthew Lopper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34, 2015.

[70] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 11 2017.

[71] Zhengyao Lv, Xiaoming Li, Xin Li, Fu Li, Tianwei Lin, Dongliang He, and Wangmeng Zuo. Learning semantic person image generation by region-adaptive normalization. *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 4 2021.

[72] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[73] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

[74] Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. Controllable person image synthesis with attribute-decomposed gan. *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 3 2020.

[75] Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Ekaterina Deyneka, Tsai-Shien Chen, Anil Kag, Yuwei Fang, Aleksei Stoliar, Elisa Ricci, Jian Ren, and Sergey Tulyakov. Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. *CVPR*, 2 2024.

[76] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *Arxiv preprint:1411.1784*, 2014.

[77] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *Arxiv preprint 2302.08453*, 2 2023.

[78] MSCOCO. https://cocodataset.org/, 2017.

[79] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *Proceedings of Machine Learning Research*, 2021.

[80] OpenPose. https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/md_doc_02_output.html, 2020.

[81] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with

spatially-adaptive normalization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[82] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision*, 12 2022.

[83] Justin Pinkney. Stable diffusion image variations. *https://github.com/justinpinkney/stable-diffusion*, 2022.

[84] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[85] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. 11 2021.

[86] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning (ICML)*, 2 2021.

[87] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations (ICLR)*, 11 2015.

[88] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *Preprint*, 2018.

[89] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *Arxiv Preprint: 2204.06125*, 4 2022.

[90] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *International Conference on Machine Learning (ICML)*, 2021.

[91] Yurui Ren, Xiaoqing Fan, Ge Li, Shan Liu, and Thomas H. Li. Neural texture extraction and distribution for controllable person image synthesis. *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 4 2022.

[92] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H. Li, and Ge Li. Deep image spatial transformation for person image generation. *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 3 2020.

[93] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR.

[94] Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

[95] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 12 2022.

[96] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer Assisted Interventions (MICCAI)*, 5 2015.

[97] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman.

Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *Arxiv preprint 2208.12242*, 8 2022.

[98] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[99] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *Arxiv preprint: 2205.11487*, 5 2022.

[100] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023.

[101] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv:1606.03498*, 2016.

[102] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[103] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *Association for Computational Linguistics (ACL)*, 2015.

[104] Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. *arXiv preprint arXiv:2304.03411*, 2023.

[105] Aliaksandr Siarohin, Enver Sangineto, Stephane Lathuiliere, and Nicu Sebe. Deformable gans for pose-based human image generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[106] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 9 2014.

[107] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[108] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021.

[109] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[110] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[111] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[112] Stability.ai. Stable diffusion 2. *https://github.com/Stability-AI/stablediffusion*, 2023.

[113] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. Df-gan: A simple and effective baseline for text-to-image synthesis. *IEEE/CVF Conference on Computer*

Vision and Pattern Recognition (CVPR), 8 2020.

[114] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, 2020.

[115] Yee Whye Teh, Max Welling, Simon Osindero, and Geoffrey E. Hinton. Energy-based models for sparse overcomplete representations. *J. Mach. Learn. Res.*, 4(null):1235–1260, Dec. 2003.

[116] Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[117] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[118] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. In *Conference on Nueral Information Processing Systems.*, 2016.

[119] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[120] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[121] Jiajun Wang, Morteza Ghahremani, Yitong Li, Björn Ommer, and Christian Wachinger. Stable-pose: Leveraging transformers for pose-guided text-to-image generation, 2024.

[122] Phil Wang. Dalle-pytorch, 2021.

[123] Qixun Wang, Xu Bai, Haofan Wang, Zekui Qin, Anthony Chen, Huaxia Li, Xu Tang, and Yao Hu. Instantid: Zero-shot identity-preserving generation in seconds. *arXiv preprint arXiv:2401.07519*, 1 2024.

[124] Tan Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. Disco: Disentangled control for referring human dance generation in real world. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2023.

[125] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[126] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems and Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003.

[127] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *SIGGRAPCH Conference Proceedings*, 12 2024.

[128] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *Arxiv Preprint 2104.14806*, 4 2021.

[129] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. *European Conference on Computer Vision (ECCV)*, 2021.

[130] Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*,

6 2024.

[131] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[132] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. *arxiv:2311.16498*, 11 2023.

[133] Lingbo Yang, Pan Wang, Chang Liu, Zhanning Gao, Peiran Ren, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Xiansheng Hua, and Wen Gao. Towards fine-grained human pose transfer with detail replenishing network. *IEEE Transactions on Image Processing*, 2020.

[134] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24)*, page 12, New York, NY, USA, 2024. ACM.

[135] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[136] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *Arxiv Pre-print 2308.06721*, 8 2023.

[137] Xiangchen Yin, Donglin Di, Lei Fan, Hao Li, Chen Wei, Xiaofei Gou, Yang Song, Xiao Sun, and Xun Yang. Grpose: Learning graph relations for human image generation with pose priors, 2024.

[138] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[139] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *International Conference on Computer Vision (ICCV)*, 2017.

[140] Jinsong Zhang, Kun Li, Yu-Kun Lai, and Jingyu Yang. Pise: Person image synthesis and editing with decoupled gan. *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 3 2021.

[141] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3d human-object spatial arrangements from a single image in the wild. *European Conference on Computer Vision (ECCV)*, 7 2020.

[142] Kaiduo Zhang, Muyi Sun, Jianxin Sun, Binghao Zhao, Kunbo Zhang, Zhenan Sun, and Tieniu Tan. Humandiffusion: a coarse-to-fine alignment diffusion framework for controllable text-driven person image generation. *Arxiv Preprint 2211.06235*, 11 2022.

[143] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *International Computer Vision Conference (ICCV)*, 2 2023.

[144] Pengze Zhang, Lingxiao Yang, Jianhuang Lai, and Xiaohua Xie. Exploring dual-task correlation for pose guided person image generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3 2022.

[145] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 1 2018.

[146] Zhenghao Zhang, Junchao Liao, Menghao Li, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. *arXiv preprint arXiv:2407.21705*, 2024.

[147] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *NeurIPS*, 5 2023.

[148] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, March 2024.

[149] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37, 2018.

[150] Xinyue Zhou, Mingyu Yin, Xinyuan Chen, Li Sun, Changxin Gao, and Qingli Li. Cross attention based style distribution for controllable person image synthesis. *European Conference on Computer Vision (ECCV) IEEE Conference of Computer Vision and Pattern Rec*, 8 2022.

[151] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Conference on Nueral Information Processing Systems.*, NIPS'17, page 465–476, 2017.

[152] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, 2017.

[153] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4 2019.

[154] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[155] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. *IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, 4 2019.

[156] Ziwei, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Liu Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.