Applications of Diffusion Probabilistic Models in Image Editing

*A paper-reading note

Wei Deng

dw-dengwei@outlook.com

***Abstract*—This article is written when reading papers about image editing using DPMs and I can practice my pooooor English.**

***Index Terms*—diffusion probabilistic models, image editing, image translation**

CONTENTS

I. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations [1]

This paper introduces a method called SDEdit, which uses different levels of details of images as input, and a pre-trained diffusion model as prior. The input can be stroke-only images, stroke-real images, and real-real images (see Fig. 1).

As well known, diffusion models employ a forward process that adds noise to the image, and a reverse process that removes the noise from the noisy images. In the forward process, it removes the high-frequency information in the early stages, and removes the low-frequency information in the later stages. In the reverse process, it first constructs the low-frequency structure and then adds the high-frequency details. Based on this principle, SDEdit first adds noise to destroy the details of the input image. This reserves the structures of the input images. Then, the reverse process starts to add details using the prior model. The full algorithm is illustrated in Fig. 2.

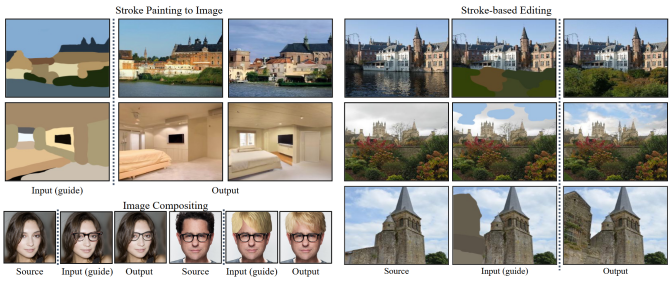

Fig. 1. SDEdit can use different levels of details of images as input.

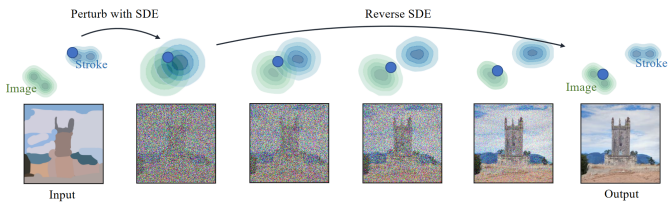

Fig. 2. Synthesizing images from strokes with SDEdit. The blue dots illustrate the editing process of our method. The green and blue contour plots represent the distributions of images and stroke paintings, respectively.

II. Prompt-to-Prompt Image Editing with Cross Attention Control [2]

In recent years, Stable Diffusion [10], an open-source diffusion model, has been widely used in many research. It can synthesize images following the input prompt. This text-to-image synthesis ability is based on classifier-free guidance. Specifically, Stable Diffusion models the joint distribution of image-text pairs via the cross-attention mechanism. The deep

Fig. 3. Cross-attention maps of a text-conditioned diffusion image generation. It displays the average attention masks for each word in the prompt that synthesized the image on the left.

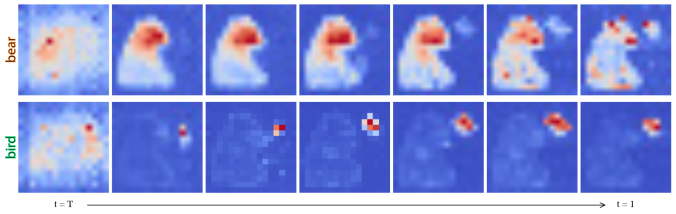

Fig. 4. The attention maps from different diffusion steps w.r.t. the words "bear" and "bird".

spatial features of the noisy image $\phi(z_t)$ are projected to a query matrix $Q = L_Q^T \phi(z_t)$, and the textual embedding is projected to a key matrix $K = L_K^T \psi(\mathcal{P})$ and a value matrix $V = L_V^T \psi(\mathcal{P})$, via learnable linear projections L_Q, L_K, and L_V. The cross-attention map is then calculated via:

$$M = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right), \tag{1}$$

where the cell $M_{i,j}$ defines the similarity between the j-th token and the i-th pixel. Finally, the cross-attention output is defined as the average of the V weighted by M, $\phi'(z_t) = MV$. The cross-attention map can be viewed from two perspectives: 1) For the i-th row of M (i.e., $M_{i,:}$), it guides the network "which token to look more" and "which token to look less", when synthesizing the i-th pixel. 2) For the j-th column of M (i.e., $M_{:,j}$), it defines how much each pixel is affected by the i-th token (see Fig. 3).

Considering two denoising routes with the same random seed: 1) Synthesizing an image corresponds to the prompt of P, and 2) Synthesizing an image corresponds to the prompt of P^*. For simplicity, we assume that P and P^* only have one different token, denoted as [DIFF] in P^*. Each route will yield cross-attention maps at each timestep, and we denote them as M_t and M_t^* respectively. What if we replace the cross-attention map P^* with P in route 2)? The synthesizing process will follow the cross-attention map P. More specifically, the token [DIFF] will only influence the pixels where P emphasizes rather than P^* emphasizes. Because the cross-attention maps determine the structure in the early steps (see Fig. 4), the structure will follow the image synthesized conditioned on P, and the appearance of the [DIFF] token attend on will follow [DIFF].

Comments: P2P can be viewed as repainting the region that the interested token provides with the target token.

III. DiffusionCLIP: Text-Guided Diffusion Models for Robust Image Manipulation [3]

This paper fine-tuned a pre-trained diffusion model via CLIP [11]: a global target loss, and a local directional loss:

$$\mathcal{L}_{\text{global}}(\mathbf{x}_{\text{gen}}, y_{\text{tar}}) = D_{\text{CLIP}}(\mathbf{x}_{\text{gen}}, y_{\text{tar}}), \tag{2}$$

where y_{tar} is a text description of the target, $\mathbf{x}_{\text{gen}}$ denotes the generated images, and $D_{\text{CLIP}}(\cdot,\cdot)$ returns the cosine distance between the inputs in the CLIP space. Another loss, the local direction loss, is designed to alleviate the issues of the global CLIP loss such as low diversity and susceptibility to adversarial attacks [TODO]. It induces the editing direction between the embedding of the generated and reference images to be aligned with the direction between the embeddings of the target and original texts:

$$\mathcal{L}_{\text{direction}}(\mathbf{x}_{\text{gen}}, y_{\text{tar}}, \mathbf{x}_{\text{ref}}, y_{\text{ref}}) = 1 - \frac{\langle \Delta I, \Delta T \rangle}{\|\Delta I\| \|\Delta T\|}, \tag{3}$$

where

$$\Delta T = \text{CLIP}_{\text{T}}(y_{\text{tar}}) - \text{CLIP}_{\text{T}}(y_{\text{ref}}), \tag{4}$$

$$\Delta I = \text{CLIP}_{\text{I}}(\mathbf{x}_{\text{gen}}) - \text{CLIP}_{\text{I}}(\mathbf{x}_{\text{ref}}), \tag{5}$$

where CLIP_{T} and CLIP_{I} are CLIP text encoder and image encoder.

The overview of DiffusionCLIP is illustrated in Fig. 5. 1) The input image $\mathbf{x}_0$ is first inversed to the latent code $\mathbf{x}_{t_0}$ via reversed DDIM, a deterministic sampling algorithm. 2) Then, DiffusionCLIP starts to fine-tune the pre-trained diffusion model ϵ_θ via the direction loss and identity loss:

$$\mathcal{L}_{\text{direction}}\left(\hat{\mathbf{x}}_0, y_{\text{tar}}; \mathbf{x}_0, y_{\text{ref}}\right) + \mathcal{L}_{\text{id}}\left(\hat{\mathbf{x}}_0, \mathbf{x}_0\right), \tag{6}$$

where

$$\mathcal{L}_{\text{id}}\left(\hat{\mathbf{x}}_0, \mathbf{x}_0\right) = \lambda_{\text{L1}} \|\mathbf{x}_0 - \hat{\mathbf{x}}_0\| + \lambda_{\text{face}} \mathcal{L}_{\text{face}}(\mathbf{x}_0, \hat{\mathbf{x}}_0). \tag{7}$$

The gradient flow can be visualized as Fig. 6. The generation algorithm is DDPM.

Comments: 1) A fine-tuned model using the DiffusionCLIP algorithm seems unable to be generalized to other tasks, such as fine-tuning with "A happy face" but wishing to synthesize "A angry face". 2) The DDIM inversion may fail when facing out-of-distribution images.

IV. Pluralistic Aging Diffusion Autoencoder [4]

This work is based on Diffusion Autoencoders (DiffAE) (see Sec. V). DiffAE views the pre-trained diffusion prior as a decoder and introduces a semantic encoder. The semantic encoder converts an image to a latent variable $\text{z} \in \mathbb{R}^5 12$ and the generated latent variable is fed into a diffusion model to reconstruct the input images.

This work, DiffAE, manipulates the latent variable with age information to achieve face aging. Specifically, 1) PADA first encodes a text-image pair using CLIP, such as "man's face in his thirties" and the associated image. They argue that the text feature contains rich age-related information (*Comments: and identity-irrelevant*). 2) Then, they propose the Probabilistic

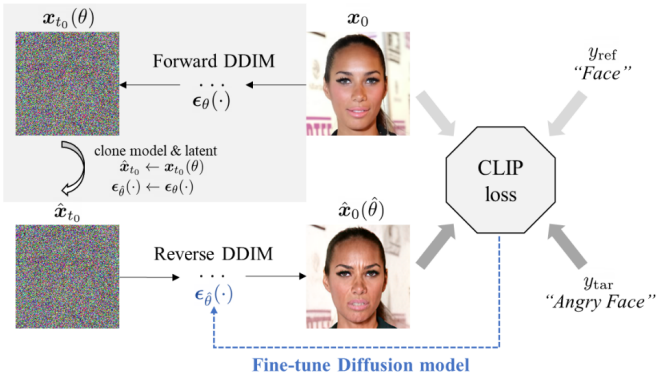

Fig. 5. Overview of DiffusionCLIP.

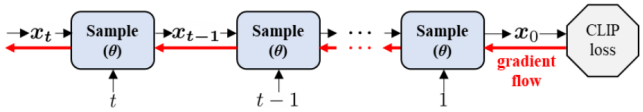

Fig. 6. Gradient flows while fine-tuning the diffusion model with the losses.

Aging Embedding, which yields a random variable in Gaussian distribution:

$$\mathcal{L}_{\text{KL}} = \text{KL}\left(\mathcal{N}(e^{\text{txt}}, I) \| \mathcal{N}(e^{\text{age}}; \mu, \sigma^2 I)\right), \tag{8}$$

where

$$e^{\text{age}} \sim \mathcal{N}(\mu, \sigma^2 I), \tag{9}$$

and μ and σ are the mean and variance vectors yielded from the Probabilistic Aging Embedding network (see the orange region in Fig. 7). 3) Finally, they transfer the aging embedding e^{age} into the space defined by the semantic encoder using an MLP-based network via:

$$\Delta z^{\text{age}} = \text{MLP}(e^{\text{age}}), \tag{10}$$

and fuse with the semantic vector of the input image:

$$z^{\text{age}} = \Gamma(\Delta z^{\text{age}}; \theta_1)\frac{z^{\text{src}} - \mu^{\text{src}}}{\sigma^{\text{src}}} + \Gamma(\Delta z^{\text{age}}; \theta_2). \tag{11}$$

Comments: This module may inspired by AdaIN.

There are age fidelity loss (like triplet loss), identity loss, normalization loss, and reconstruction loss.

Comments: I argue that this paper's contribution is injecting age information into the semantic vector and using text and a reference image to provide the age information, which implicitly disentangled age and identity. The whole training process does not need image pairs including the same identity and different ages. I have not completely read the supplementary materials. TODO

V. Diffusion Autoencoders: Toward a Meaningful and Decodable Representation [5]

Diffusion models have recently succeeded in synthesizing realistic and high-resolution images. The authors questioned whether diffusion models can serve as a good representation-learner. Although DDIM inversion can convert an input image x_0 into a spatial latent variable x_T by its deterministic process, the generated latent variable lacks high-level semantics and other properties, such as disentanglement, and meaningful interpolation in the latent space (see Fig. 8).

GAN inversion, which optimizes the latent code to reconstruct the given input. Even though the resulting code carries rich semantics, this technique struggles to faithfully reconstruct the input image since there exists an information bottleneck in the latent code. This work aims to find a decodable meaningful representation, which requires capturing both high-level semantics and low-level stochastic variations. The key idea is to learn high-level semantic representations using a learnable encoder and low-level stochastic variations using a diffusion model. Specifically, they trained a semantic encoder and a decoder. The decoder is a conditional variant of DPMs, and it takes two latent codes as input. 1) The first subcode is the semantic code, which is converted from the input image by the semantic encoder. 2) The second subcode is the stochastic code, which is inferred by reversing the DDIM process conditioned on the semantic code.

A. DDIM inversion

DDIM is a deterministic sampling algorithm:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}}\right)\epsilon_\theta^t(\mathbf{x}_t), \tag{12}$$

which can be viewed as an ordinary differential equation (ODE) by rewriting to:

$$\frac{\mathbf{x}_{t-1}}{\sqrt{\alpha_{t-1}}} - \frac{\mathbf{x}_t}{\sqrt{\alpha_t}} = \left(\sqrt{1-\alpha_{t-1}} - \sqrt{1-\alpha_t}\right)\epsilon_\theta^t(\mathbf{x}_t). \tag{13}$$

The forward DDIM process (DDIM inversion) to obtain the latent is represented as:

$$\mathbf{x}_t = \sqrt{\frac{\alpha_t}{\alpha_{t-1}}}\frac{\mathbf{x}_{t-1}}{\epsilon_\theta^t(\mathbf{x}_t)} - \sqrt{\alpha_t(1-\alpha_{t-1})} + \sqrt{1-\alpha_t}. \tag{14}$$

B. Semantic Encoder

The goal of the semantic encoder $\mathbf{z}_{\text{sem}} = \text{Enc}(\mathbf{x}_0)$ is to summarize the input image into a descriptive vector. Here, $\mathbf{z}_{\text{sem}}$ is a non-spatial vector of dimension $d = 512$ and resembles the style vector in StyleGAN.

C. Stochastic Encoder

The output of the stochastic encoder $\mathbf{x}^T = \text{DDIM\_inv}(\mathbf{x}_0, \mathbf{z}_{\text{sem}})$ is encouraged to encode only the information left out by $\mathbf{z}_{\text{sem}}$. It contains the low-frequency information of the input image and is a spatial tensor with size $C \times H \times W$.

D. Diffusion Autoencoders

The encoders first convert the input image into latent codes $\mathbf{z} = (\mathbf{z}_{\text{sem}}, \mathbf{x}^T)$. Then, DiffAE uses a conditional diffusion model as a decoder, which reconstructs the input image from the latent codes with DDIM sampling:

$$p_\theta(\mathbf{x}_{0:T}|\mathbf{z}_{\text{sem}}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_{\text{sem}}). \tag{15}$$

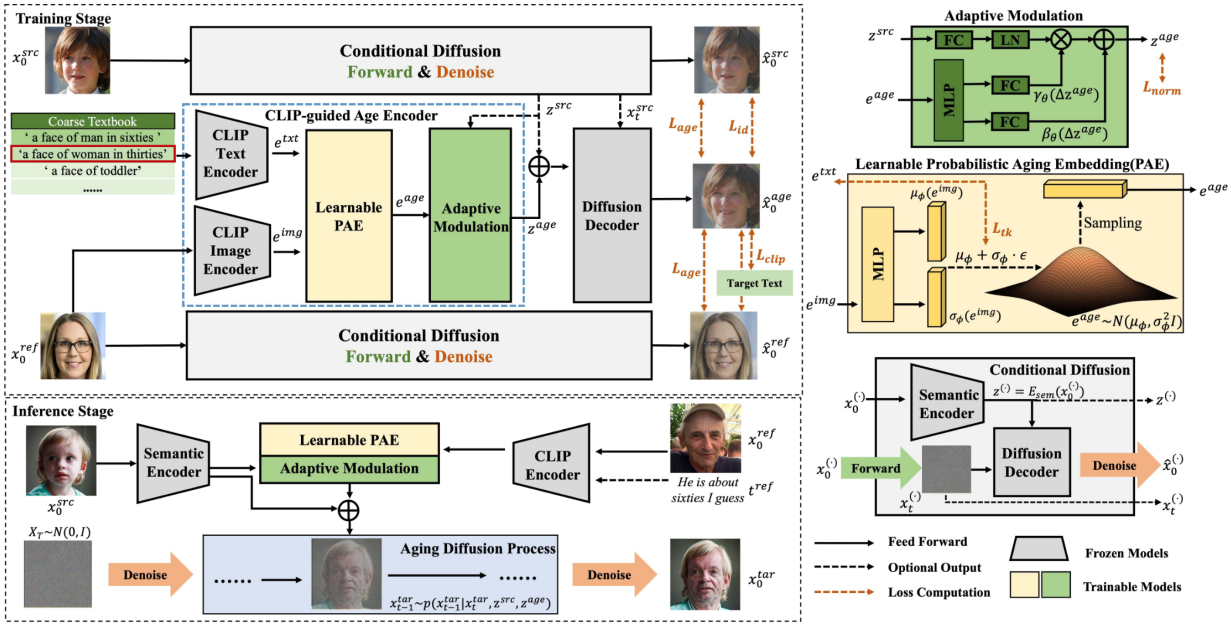

Fig. 7. Overview of PADA

Fig. 8. Interpolation between two real images in the latent space via DDIM inversion. DDIM inversion produces non-smooth transitions.

The optimizing object is to minimize the simple loss function defined in DDPM:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0\sim p_{\text{data}},\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim[1,T]}\left[\|\epsilon_\theta(\mathbf{x}_t,t,\mathbf{z}_{\text{sem}})-\epsilon\|_2^2\right] \tag{16}$$

E. Unconditional Sampling

The major challenge of unconditional sampling of DiffAE is how to sample $\mathbf{z}_{\text{sem}}$. The distribution of $\mathbf{z}_{\text{sem}}$ belongs to is a complex distribution. In this work, the authors trained another diffusion model to generate $\mathbf{z}_{\text{sem}}$. Unlike image diffusion models, the architecture of the diffusion model synthesizing $\mathbf{z}_{\text{sem}}$ is a deep MLP with skip connections (UNet with transformer blocks is used in image diffusion models), and the loss function is L-1 loss (L-2 loss function is used in image diffusion models).

F. Summary

1) The diffusion autoencoder can achieve meaningful latent interpolation, and high-quality image reconstruction thanks to the utilization of low-level stochastic and high-level semantic latent codes. 2) Because of the rich information about $\mathbf{x}_0$ in $\mathbf{z}_{\text{sem}}$, it can guide the sampling algorithm to reach $\mathbf{x}_0$ resulting in faster synthesizing compared with the models without latent semantic codes. 3) The distribution where $\mathbf{z}_{\text{sem}}$ from is meaningful. It can perform image manipulation via moving $\mathbf{z}_{\text{sem}}$ in the latent space.

VI. DiffMorpher: Unleashing the Capability of Diffusion Models for Image Morphing [6]

Diffusion models have achieved remarkable image generation quality. However, compared to GANs, they have difficulties in smooth interpolation between two images 8. Such smooth interpolation is intriguing as it naturally serves as a solution for the image morphing task with many applications [12]. The key idea of this work is to fit two LoRAs with two real images and interpolate between both the LoRA parameters and the DDIM latent variables.

A. LoRA Interpolation

Low-Rank Adaption (LoRA) is an efficient tuning technique that was first proposed to fine-tune large language models and recently introduced to the domain of diffusion models. Instead of directly tuning the entire model, LoRA fine-tunes the model parameters θ by training a low-rank residual part $\Delta\theta$. The authors found that LoRA enjoys an impressive capacity to encapsulate high-level semantics into the low-rank parameter space. By fitting a LoRA on a real image, the fine-tuned model can generate diverse samples with consistent semantic identity when traversing the latent noise, see Fig. 9. They fit two LoRAs ($\Delta\theta_0$ and $\Delta\theta_1$) on two real images and interpolation between the LoRAs:

$$\Delta\theta_\alpha = (1-\alpha)\Delta\theta_0 + \alpha\Delta\theta_1 \tag{17}$$

B. Latent Interpolation

After LoRA interpolation, the next step is to find the algorithm to interpolate the latent noise $\mathbf{z}_{T\alpha}$ and the latent condition $\mathbf{c}_\alpha$. Here, they first use DDIM inversion to obtain the

Fig. 9. A LoRA fitted to an image tends to capture its semantic identity, while the layout and appearance are controlled by latent noise.

latent noise (denoted as $\mathbf{z}_{T0}$ and $\mathbf{z}_{T1}$) of the correspondence input images. Then, the intermediate noise $\mathbf{z}_{T\alpha}$ is obtained through "slerp" algorithm:

$$\mathbf{z}_{T\alpha} = \frac{\sin(\phi(1-\alpha))}{\sin\phi}\mathbf{z}_{T0} + \frac{\sin(\phi\alpha)}{\sin\phi}\mathbf{z}_{T1}, \tag{18}$$

where $\phi = \arccos\left(\frac{\mathbf{z}_{T0}^T\mathbf{z}_{T1}}{\|\mathbf{z}_{T0}\|\|\mathbf{z}_{T1}\|}\right)$. It is worth noting that the vanilla DDIM inversion is known to suffer from unfaithful reconstruction. Hence, the latent variables are obtained via the LoRA-based fine-tuned UNet models. Regarding the interpolation of the condition embeddings, they found that linear interpolation between **aligned** input conditions can serve as meaningful intermediate conditions:

$$\mathbf{c}_\alpha = (1-\alpha)\mathbf{c}_0 + \alpha\mathbf{c}_1. \tag{19}$$

C. Self-Attention Interpolation and Replacement

Just utilizing LoRA interpolation and latent interpolation still fails to achieve smooth changes. Inspired by attention-base diffusion editing methods (such as prompt-to-prompt, pix2pix-zero, etc) This paper proposed a method named self-attention interpolation and replacement. As shown in Fig. 10, they interpolate the Ks and Vs matrices at each self-attention module and replace the corresponding matrices in the intermediate LoRA. In particular, replacing attention features in all denoising steps may lead to blurred images. Thus, they only replace the features at the early λT ($\lambda \in (0,1)$) steps (generate low-level structures) and leave the self-attention modules unchanged in the remaining steps.

D. AdaIN Adjustment

To ensure the coherence in color and brightness between the generated images and the input images, they introduced the Adaptive Instance Normalization (AdaIN) adjustment. They calculate the mean μ_i and the standard deviation σ_i ($i = \{0,1\}$) for each channel of the latent noises, and interpolate between μ_i and σ_i as the adjustment target of the intermediate noises:

$$\begin{aligned}
\mu_\alpha &= (1-\alpha)\mu_0 + \alpha\mu_1 \\
\sigma_\alpha &= (1-\alpha)\sigma_0 + \alpha\sigma_1 \\
\tilde{\mathbf{z}}_{0\alpha} &= \sigma_\alpha\left(\frac{\mathbf{z}_{0\alpha} - \mu(\mathbf{z}_{0\alpha})}{\sigma(\mathbf{z}_{0\alpha})}\right) + \mu_\alpha.
\end{aligned} \tag{20}$$

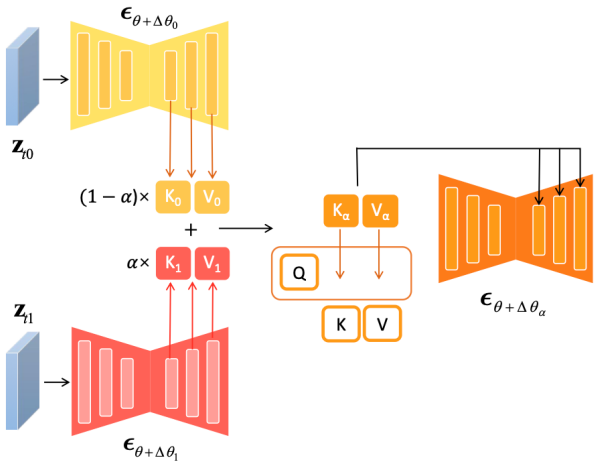

Fig. 10. Self-Attention Interpolation and Replacement

Fig. 11. (a) DDIM baseline, (b) + LoRA interpolation, (c) + attention interpolation and replacement, (d) + reschedule (DiffMorpher).

E. Reschedule Sampling

Furthermore, they introduced a new rescheduling method. Formally, assuming $D(\mathcal{I}_i, \mathcal{I}_j)$ $(i,j \in [0,1])$ is the perceptual distance between $\mathcal{I}_i$ and $\mathcal{I}_j$. Given the number of frames n, the object is to make the variance of $\{D(\mathcal{I}_i, \mathcal{I}_{i+\frac{1}{n}}) | i \in \{0, \frac{1}{n}, \cdots, 1-\frac{1}{n}\}\}$ to be as small as possible. Define ΔD w.r.t. α:

$$\Delta D(\alpha) \triangleq \begin{cases} D(\mathcal{I}_0, \mathcal{I}_{\frac{1}{n}})/\bar{D} & \text{if } 0 \le \alpha < \frac{1}{n} \\ D(\mathcal{I}_{\frac{1}{n}}, \mathcal{I}_{\frac{2}{n}})/\bar{D} & \text{if } \frac{1}{n} \le \alpha < \frac{2}{n} \\ \cdots & \cdots \\ D(\mathcal{I}_{1-\frac{1}{n}}, \mathcal{I}_1)/\bar{D} & \text{if } 1-\frac{1}{n} \le \alpha < 1 \end{cases}, \tag{21}$$

where $\bar{D} = \sum_{i=0}^{1-\frac{1}{n}} D(\mathcal{I}_i, \mathcal{I}_{i+\frac{1}{n}})$ is the sum of perceptual distance between all **adjacent** frames.

TODO.

F. Ablation Results

See Fig. 11.

G. Diffusion Models already have a Semantic Latent Space [13]

The lack of semantic latent space which is essential for controlling the generative process. This paper found that the deepest feature map in the pre-trained diffusion model has nice properties to accommodate semantic image manipulation: 1) homogeneity, 2) linearity, 3) robustness, and 4) consistency across timesteps. They named the semantic space as *h-space*.

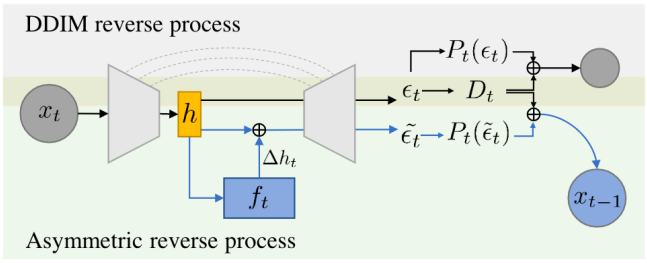

Fig. 12. Generative process of Asyrp.

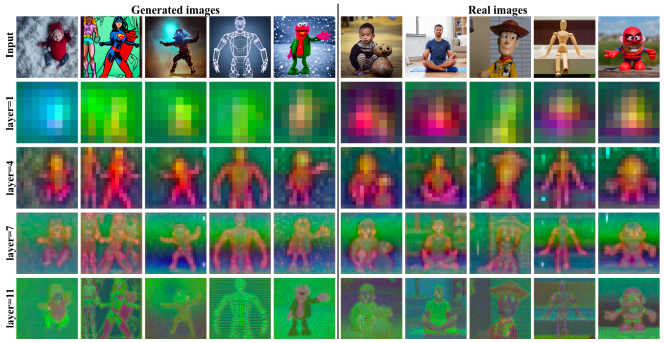

Fig. 13. Visualizing diffusion decoder features in different decoder layers. The deep features are passed to PCA and remain three primary components for visualization. It can be found that: 1) The shallow layer (layer 1) features differ from image semantics and are influenced by appearance. 2) The middle layer (layer 4) features reveal semantic regions. The same semantic regions share the same color across all images. It is not influenced by appearance. 3) The deeper layer (layers 7-11) features capture high-frequency (low-level) information. It is influenced by appearance.

Fig. 12 shows the generative process of Asyrp. It introduces a lightweight residual network f_t in the *h-space* to convert the deep feature to the feature with expected attributes via CLIP loss.

VII. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation [7]

The famous LDM model, Stable Diffusion, is based on the UNet, which includes residual blocks, self-attention layers, and cross-attention blocks. This paper found that: 1) spatial features extracted by the intermediate **decoders** contain the localized semantic information and are less affected by appearance information (see Fig. 13). 2) the self-attention representing the affinities between the spatial features, allows to retain fine layout and shape details (see Fig. 14).

Based on these observations, this paper translates an input image by overriding the decoder feature maps and self-attention maps, which contain the layout of the input image.

VIII. DeltaSpace: A Semantic-aligned Feature Space for Flexible Text-guided Image Editing [8]

Manipulating the latent code in the latent space is a common method for image editing and how to find the target latent code is the major issue in the latent-based manipulation field. Since the CLIP model connects the image and textual spaces,

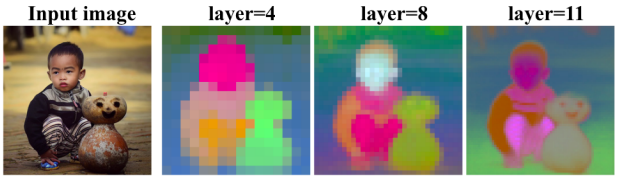

Fig. 14. Self-attention map visualization. The self-attention maps are aligned with the layout of the image: similar regions share similar colors.

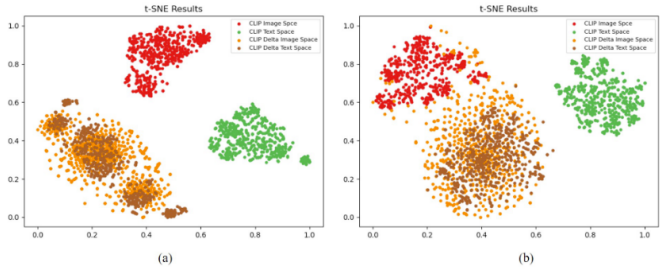

Fig. 15. Feature space analysis on (a) MultiModal-CelebA-HQ dataset and (b) MS-COCO dataset.

it is widely utilized for text-conditioned image editing. In this paper, the authors found that the features extracted by CLIP from paired text-image data are not close to each other (see Fig. 15). They randomly select 2 text-image pairs and extract their features, which are denoted as f_i^1, f_i^2, f_t^1, and f_t^2, and define:

$$\begin{aligned} \Delta t &= f_t^1 - f_t^2, \\ \Delta i &= f_i^1 - f_i^2. \end{aligned} \tag{22}$$

Then they visualize Δt and Δi as shown in Fig. 15. It can be found that the CLIP feature difference space for image and text exhibits better alignment and semantic consistency (i.e. $\Delta i \approx \Delta t$). Based on this prior, this paper builds a model that inputs the source image feature, source textual feature, and image/textual direction (i.e. $\Delta i / \Delta t$) and outputs the latent direction Δs (see Fig. 16).

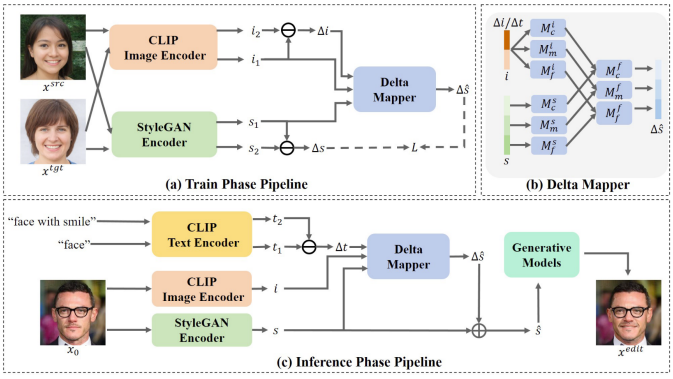

Fig. 16. The overall framework of the proposed DeltaEdit.

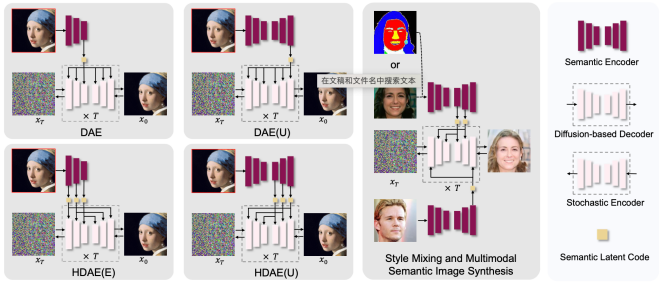

Fig. 17. Overview of different network structures.

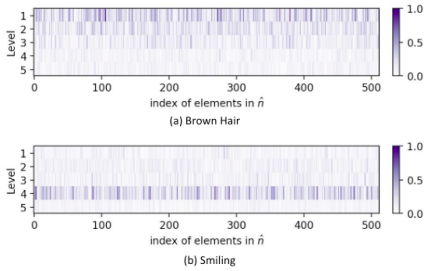

Fig. 18. The values of the hyper-planes at 5 levels.

IX. Hierarchical Diffusion Autoencoders and Disentangled Image Manipulation [9]

Probing and manipulating diffusion models' latent space has not been extensively explored. Though DiffAE converts an input image into a 512-dimension semantic vector, it neglects low-level and mid-level details. The main reason is that DiffAE only uses the deepest feature of the UNet model as the semantic vector (see DAE in Fig. 17). Thus, this paper projects multi-level features into 512-dimension vectors as semantic vectors (see HDAE(E) and HDAE(D) in Fig. 17). Moreover, they trained linear classifiers and found that most elements of the hyper-planes are nearly 0 (see Fig. 18). This indicates that the few high-value elements indicate the dominant direction of an attribute classifier, while the majority, of low-value elements are noisy and may lead to attribute entanglement. Therefore, this paper maintains the top-k values of the hyper-planes and sets others to be 0 for disentanglement.

References

[1] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Guided image synthesis and editing with stochastic differential equations," *arXiv preprint arXiv:2108.01073*, 2021.

[2] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, "Prompt-to-prompt image editing with cross attention control," *arXiv preprint arXiv:2208.01626*, 2022.

[3] G. Kim, T. Kwon, and J. C. Ye, "Diffusionclip: Text-guided diffusion models for robust image manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 2426–2435.

[4] P. Li, R. Wang, H. Huang, R. He, and Z. He, "Pluralistic aging diffusion autoencoder," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 22 613–22 623.

[5] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn, "Diffusion autoencoders: Toward a meaningful and decodable representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 619–10 629.

[6] K. Zhang, Y. Zhou, X. Xu, X. Pan, and B. Dai, "Diffmorpher: Unleashing the capability of diffusion models for image morphing," *arXiv preprint arXiv:2312.07409*, 2023.

[7] N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel, "Plug-and-play diffusion features for text-driven image-to-image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1921–1930.

[8] Y. Lyu, K. Zhao, B. Peng, Y. Jiang, Y. Zhang, and J. Dong, "Deltaspace: A semantic-aligned feature space for flexible text-guided image editing," *arXiv preprint arXiv:2310.08785*, 2023.

[9] Z. Lu, C. Wu, X. Chen, Y. Wang, L. Bai, Y. Qiao, and X. Liu, "Hierarchical diffusion autoencoders and disentangled image manipulation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2024, pp. 5374–5383.

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html

[12] A. Q. Aloraibi, "Image morphing techniques: A review," *Technium: Romanian Journal of Applied Sciences and Technology*, vol. 9, p. 41–53, Apr. 2023. [Online]. Available: https://www.techniumscience.com/index.php/technium/article/view/8699

[13] M. Kwon, J. Jeong, and Y. Uh, "Diffusion models already have a semantic latent space," *arXiv preprint arXiv:2210.10960*, 2022.