X-Adapter: Adding Universal Compatibility of Plugins for Upgraded Diffusion Model

Lingmin Ran[1] Xiaodong Cun[3] Jia-Wei Liu[1] Rui Zhao[1] Song Zijie[4]
Xintao Wang[3] Jussi Keppo[2] Mike Zheng Shou[1]

[1]Showlab, [2]National University of Singapore [3]Tencent AI Lab [4]Fudan University

Figure 1. Given pretrained plug-and-play modules (*e.g.*, ControlNet, LoRA) of the base diffusion model (*e.g.*, Stable Diffusion 1.5), the proposed X-Adapter can universally upgrade these plugins, enabling them directly work with the upgraded Model (*e.g.*, SDXL) without further retraining. Text prompts: *"1girl, solo, smile, looking at viewer, holding flowers" "Apply face paint" "1girl, upper body, flowers" "A cute cat holding a gun" "Best quality, extremely detailed" "A fox made of water"* from left to right, top to bottom.

Abstract

*We introduce X-Adapter, a universal upgrader to enable the pretrained plug-and-play modules (*e.g.*, ControlNet, LoRA) to work directly with the upgraded text-to-image diffusion model (*e.g.*, *SDXL) without further retraining. We achieve this goal by training an additional network to control the frozen upgraded model with the new text-image data pairs. In detail, X-Adapter keeps a frozen copy of the old model to preserve the connectors of different plugins. Additionally, X-Adapter adds trainable mapping layers that bridge the decoders from models of different versions for feature remapping. The remapped features will be used as guidance for the upgraded model. To enhance the guidance ability of X-Adapter, we employ a null-text training strategy for the upgraded model. After training, we also introduce a two-stage denoising strategy to align the initial latents of X-Adapter and the upgraded model. Thanks to our strategies, X-Adapter demonstrates universal compatibility with various plugins and also enables plugins of different versions to work together, thereby expanding the functionalities of diffusion community. To verify the effectiveness of the pro-*

posed method, we conduct extensive experiments and the results show that X-Adapter may facilitate wider application in the upgraded foundational diffusion model. Codes and models will be made available.

1. Introduction

Large text-to-image diffusion models [29, 32, 35] have drawn the attention of both researchers and creators nowadays. Since these models are often trained on thousands of GPU days with millions of data pairs, the major development of the current research focuses on designing plug-and-play modules [12, 17, 24, 43], which are commonly called plugins, to add new abilities on the pre-trained text-to-image models. People use plugins for photo creation [17, 34, 39], controllable drawing [24, 43], and editing [22, 30], both for image and video [9, 12, 42]. The development speed of downstream plugins is faster than the release of the base model since it is easier to train and enables many more different features. But when a larger foundation model (*e.g.*, SDXL [29]) is released, all the downstream plugins need to be retrained for this upgraded model, which takes much more time for maintenance and upgradation.

We aim to solve this inconvenient plugin incompatibility when upgradation by proposing a unified adapter network, where all the downstream plugins in the original base model (*e.g.*, Stable Diffusion v1.5 [32]) can be directly used in upgraded model (*e.g.*, SDXL [29]) via the proposed method. However, this task has a lot of difficulties. First, when training different diffusion model versions, the compatibility of plugins is often not considered. Thus, the original connector of the plugin might not exist in the newly upgraded model due to dimension mismatch. Second, different plugins are applied in the different positions of the Stable Diffusion. For example, ControlNet [43] and T2I-Adapter [24] are added at the encoder and decoder of the fixed denoising UNet respectively. LoRA [17] are added after each linear layer of a fixed denoising UNet. This uncertainty makes it difficult to design a unified plugin. Finally, although most current models are based on the latent diffusion model [32], the latent space of each model is different. This gap is further boosted between the diffusion models in pixel and latent space.

We propose X-Adapter to handle above difficulites. In detail, inspired by ControlNet [43], we consider X-Adapter as an additional controller of the upgraded model. To solve the problem of the connector and the position of different plugins, we keep a frozen copy of the base model in the X-Adapter. Besides, we design several mapping layers between the decoder of the upgraded model and X-Adapter for feature remapping. In training, we only train the mapping layers concerning the upgraded model without any plugins. Since the base model in the X-Adapter is fixed, the old plugins can be inserted into the frozen diffusion model copy in the X-Adapter. After training, we can sample two latent for X-Adapter and an upgraded model for inference. To further boost the performance, we also propose a two-stage inference pipeline by sequentially inference Stable Diffusion v1.5 first and then the SDXL inspired by SDEdit [22]. Experiments show that the proposed method can successfully upgrade the plugins for larger models without specific retraining. We also conduct numerical experiments to show the effectiveness of two widely used plugins, *i.e.*, ControlNet [43], and LoRA [17].

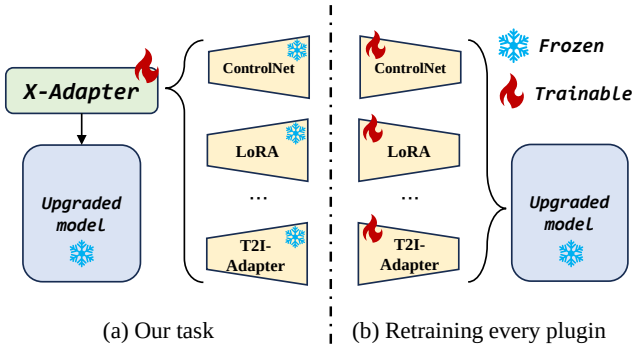
In summary, the contribution of this paper can be summarized as:

- We target a new task in the large generative model era where we need to update plugins for different foundational models.
- We propose a general framework to enable upgraded model compatible with pretrained plugins. We propose a novel training strategy that utilizes two different latent with mapping layers. Besides, we design two kinds of inference strategies to further boost the performance.
- Experiments show the proposed methods can successfully make old plugins work on upgraded text-to-image model with better performance compared to the old foundational model.

2. Related Works

Diffusion Model for Text-to-Image Generation. Diffusion models are initially proposed by Sohl-Dickstein et al. [36], and have recently been adapted for image synthesis [10, 19]. Beyond unconditional image generation, the text-to-image diffusion models [32] is an important branch of the image diffusion model, since it leverages larger-scale datasets for training. In these networks, Glide [25] proposes a transformer [37] based network structure. Imagen [35] further proposes a pixel-level cascaded diffusion model to generate high-quality images. Different from pixel-level diffusion, the technique of Latent Diffusion Models (LDM) [32] conducts diffusion in a latent image space [18], which largely reduces computational demands. Stable Diffusion v1.5 [7] is a large-scale pre-trained latent diffusion model. Stable Diffusion v2.1 [8] and SDXL [29] are the following versions of Stable Diffusion v1.5 by optimizing latent space, network structure, and training data. Compared to Midjourney [23] and DALL [26, 27], SDXL achieves state-of-the-art results.

Plugins for Text-to-Image Diffusion Model. Since the stable diffusion model [32] is open-sourced, plug-and-play modules, commonly referred to as "plugins", significantly expand the capabilities of pre-trained text-to-image (T2I) models. GLIGEN [20] adds an additional gate attention for grounded generation. LoRA [17] is a general parameter-efficient training method that allows us to fine-tune the



(a) Our task (b) Retraining every plugin

Figure 2. ***Task Definition.*** Different from the previous method to train each plugin individually, our method only trains a single X-Adapter to all the fixed downstream plugins.

stable diffusion for stylization and customization easily. Dreambooth [34] and Textual Inversion [11, 40] customize personal concepts by finetuning the pre-trained diffusion model. IP-Adapter [39] extends these works for universal image variation. Besides, ControlNet [43] and T2I-Adapter [24] add spatial conditioning controls to diffusion models by incorporating an extra network to encode conditions. AnimateDiff [12] allows a personalized T2I model to generate videos with high temporal consistency by adding a temporal module. Although these plugins are powerful, it is unfeasible to apply an old plugin to an upgraded T2I model, which significantly hampers the development and application of diffusion models.

Parameter-Efficient Transfer Learning. Our topic is also related to parameter-efficient transfer learning since we need to remedy the domain gap when upgrading. The emergence of large-scale pre-trained models, *e.g.*, Stable Diffuions [32], CLIP [31], has highlighted the importance of the effective transfer of these foundational models to downstream tasks. Parameter-efficient Transfer Learning (PETL) methods [15, 41, 44] add additional parameters to the original model to overcome the domain gaps between the pre-trained dataset and target tasks. PMLR [15] introduces an adapter that consists of a down-sampling layer and an up-sampling layer and inserts it into Transformer [37] blocks. Zhao et al. [44] bridge the domain gap by aligning the dataset's distribution. Zhang et al. [41] propose a task-agnostic adapter among various upstream foundation models. Similar to upgrading the CLIP for visual understanding [41], our objective is to enable upgraded diffusion models compatible with all kinds of plugins.

3. Methods

3.1. Task Definition

We aim to design a universal compatible adapter (X-Adapter) so that plugins of the base stable diffusion model can be directly utilized in the upgraded diffusion model. As shown in Fig. 2, given a powerful pre-trained text-to-image diffusion model M_{new} (*i.e.*, SDXL [29]), we aim to design a universal adapter X-Adapter so that all the pre-trained down-stream plugins (*e.g.*, ControlNet [43], T2I-Adapter [24], LoRA [17]) on M_{base} (*i.e.*, Stable Diffusion v1.5 [32]) can work smoothly on M_{new} without requiring additional training. Thanks to this universal adaption, we highlight some potential benefits:

(i) Universal Compatibility of Plugins from Base Model. A naive idea to apply a plugin network to the new model is to directly train the specific downstream plugin individually. However, take ControlNet [43] family as an example, it would require training more than ten different networks to achieve the original abilities. Differently, our method only needs to train *one* version-to-version adapter in advance and enable direct integration of pre-trained plugins from the base model, *i.e.*, Stable Diffusion v1.5 [32].

(ii) Performance Gain with respect to Base Model. Since original plugins are only trained on the base model, their power is also restricted due to the limited generative capability. Differently, our adapter can improve the performance of these plugins by the upgraded models since these new models are typically more powerful in terms of visual quality and text-image alignments.

(iii) Plugin Remix Across Versions. Since we retain the weights of both the base and upgraded models, our method also enables the use of plugins from both models (*e.g.* ControlNet of Stable Diffusion v1.5 and LoRA of SDXL can work together smoothly as if ControlNet were originally trained on SDXL). It largely expands the applicability of the plugins from different development stages of the text-to-image community.

3.2. Preliminary: Latent Diffusion Model

Before introducing our method, we first introduce the Latent Diffusion Model (LDM [32]), since most of the open-source models are based on it. LDM extends denoising diffusion model [14] for high-resolution image generation from text prompt, which first uses a VAE [18]'s encoder $\mathcal{E}$ to compress the RGB image x into latent space z. After that, a UNet [33] ϵ_θ is used to remove added noise from a noisy latent. Formally, ϵ_θ is trained using the following objective:

$$\min_{\theta} E_{z_0,\epsilon\sim N(0,I),\boldsymbol{t}\sim \text{ Uniform }(1,T)} \left\|\epsilon - \epsilon_\theta\left(\boldsymbol{z}_t, \boldsymbol{t}, \boldsymbol{c}\right)\right\|_2^2, \quad (1)$$

where z_t is the noisy latent of z from timestep t and c is the embedding of conditional text prompt.

3.3. X-Adapter

X-Adapter is built upon the base Stable Diffusion v1.5 [32] to maintain the full support for the plugin's connectors. Additionally, in the decoder of each layer, we train an additional mapping network to map the features from the base model to the upgraded model (*e.g.*,

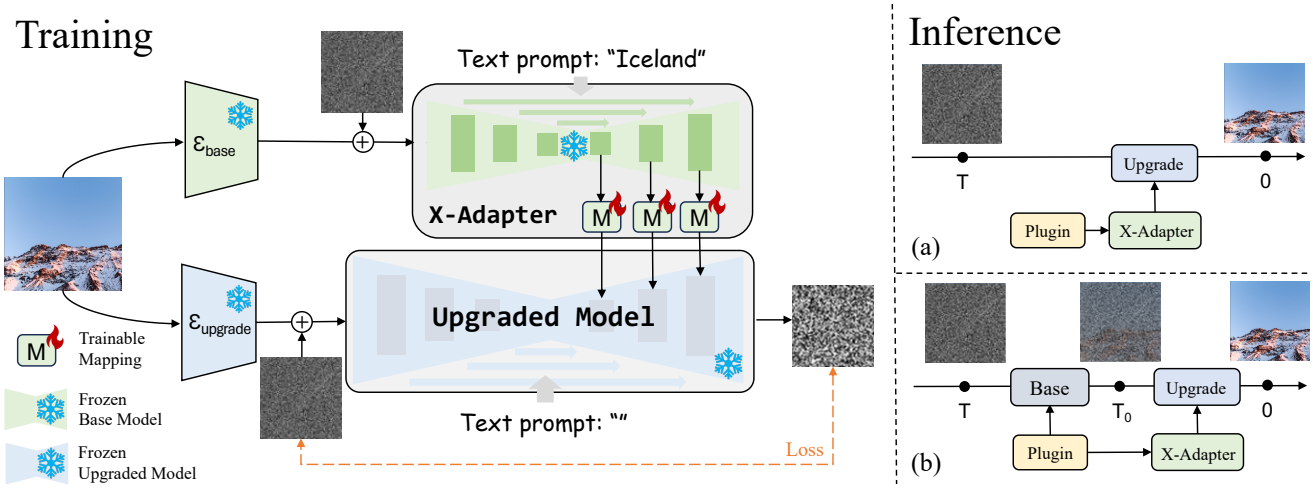


Figure 3. ***Method Overview***. In training, we add different noises to both the upgraded model and X-Adapter under the latent domain of base and upgraded model. By setting the prompt of the upgraded model to empty and training the mapping layers, X-Adapter learns to guide the upgraded model. In testing, (a) we can directly apply the plugins on the X-Adapter for the upgraded model. (b) A two-stage influence scheme is introduced to improve image quality.

SDXL [29]) for guidance as shown in Fig. 3. In each mapper, a stack of three ResNet [13] is utilized for dimension matching and feature extraction. Formally, suppose we have N adapters and $\mathcal{F}_n(\cdot)$ denotes the n^{th} trained mapper, given multi-scale feature maps $\boldsymbol{F}_{base} = \{\boldsymbol{F}^1_{base}, \boldsymbol{F}^2_{base}, ..., \boldsymbol{F}^N_{base}\}$ from base model, guidance feature $\boldsymbol{F}_{mapper} = \{\boldsymbol{F}^1_{mapper}, \boldsymbol{F}^2_{mapper}, ..., \boldsymbol{F}^N_{mapper}\}$ is formed by feeding $\boldsymbol{F}_{base}$ to the mapping layers. Note that the dimension of $\boldsymbol{F}_{mapper}$ is the same as that of certain intermediate features of upgraded decoder layers. $\boldsymbol{F}_{mapper}$ is then added with those layers. In summary, the guidance feature extraction and fusion can be defined as the following formulation:

$$\boldsymbol{F}^n_{mapper} = \mathcal{F}_n(\boldsymbol{F}^n_{base}) \quad (2)$$

$$\boldsymbol{F}^n_{up} = \boldsymbol{F}^n_{up} + \boldsymbol{F}^n_{mapper}, n \in \{1, 2, ..., N\}, \quad (3)$$

where $\boldsymbol{F}^n_{up}$ denotes upgraded model's n^{th} decoder layer to fuse guidance feature.

Training Strategy. As shown in Fig. 3, given an upgraded diffusion model, X-Adapter is firstly trained in a plugin-free manner on the upgraded diffusion model for text-to-image generation. Formally, given an input image $\mathcal{I}$, we first embed it to the latent spaces z_0 and $\overline{z}_0$ via base and upgraded autoencoder respectively. Then, we randomly sample a time step t from $[0, T]$, adding noise to the latent space, and produce two noisy latent z_t and $\overline{z}_t$ for denoising. Given timestep $\boldsymbol{t}$, the prompt $\boldsymbol{c}_b$ of X-Adapter and upgraded model's prompt $\boldsymbol{c}_u$, X-Adapter is trained with the upgraded diffusion network ϵ_θ to predict the added noise ϵ by:

$$E_{\overline{\boldsymbol{z}}_0, \epsilon, \boldsymbol{t}, \boldsymbol{c}_b, \boldsymbol{c}_u} \left\| \epsilon - \epsilon_\theta \left(\boldsymbol{z}_t, \boldsymbol{t}, \boldsymbol{c}_u, \mathcal{X}_{Adapter}(\overline{\boldsymbol{z}}_t, \boldsymbol{t}, \boldsymbol{c}_b) \right) \right\|_2^2 . \quad (4)$$

In the training process, the objective of the above loss function is to determine the offsets between the X-Adapter and the upgraded space. Inspired by previous task-compatibility plugins for additional control signal [24, 43] and video generation [12], we find that the key to task-agnostic adaptation is to fix the parameters of the trained diffusion UNet. Thus, we freeze the parameters in the base model during training, which ensures that old plugins can be seamlessly inserted. To avoid affecting the original high-quality feature space of the upgraded model, we also freeze its parameters similar to conditional control methods [24, 43]. All text prompts $\boldsymbol{c}_u$ are set to an empty string inspired by [21]. Thus, the upgraded model provides the average feature space with an empty prompt, while X-Adapter learns the offset given base feature space, guiding the native upgraded model. Although $\boldsymbol{c}_u$ is set to empty during training, our experiments show that we do not need to adhere this rule during inference and X-Adapter works well with any $\boldsymbol{c}_u$ after training. After training, the plugins can naturally be added to X-Adapter for their abilities.

Inference Strategy. During training, two bypasses' latents are encoded from the same image, naturally aligning with each other. However, since the latent space of the two models is different, during the inference stage, if the initial latents for two bypasses are randomly sampled (Fig. 3 (a)), this leads to a lack of alignment, potentially causing conflicts that affect the plugin's function and image quality.

To tackle this issue, inspired by SDEdit [22], we propose a two-stage inference strategy as shown in Fig. 3 (b). Given total timestep T, at the first stage, we randomly sample an initial latent z_T for X-Adapter and run with plugins in timestep T_0 where $T_0 = \alpha T, \alpha \in [0, 1]$. At timestep T_0, the base model's latent z_{T_0} will be converted to upgraded

model's latent $\overline{z}_{T_0}$ by:

$$\overline{z}_{T_0} = \mathcal{E}_{up}(\mathcal{D}_{base}(z_{T_0})), \tag{5}$$

where $\mathcal{D}_{base}$ denotes the base model's decoder and $\mathcal{E}_{up}$ denotes the upgraded model's encoder. $\overline{z}_{T_0}$ and z_{T_0} will be initial latents for two bypasses at the second stage where the plugin will guide the upgraded model's generation through X-Adapter. We observe that for most plugins our framework performs optimally when $T_0 = \frac{4}{5}T$, *i.e.*, the base model run 20% of the time step for warmup and then runs our X-Adapter in the rest of the inference time directly. We give detailed ablations on this two-stage inference in the experiments.

4. Experiments

4.1. Implementation Details

We implement X-Adapter using Stable Diffusion v1.5 [32] as the base model, and SDXL [29] base as the main upgraded model. Mapping layers of X-Adapter are placed at the base model's last three decoder blocks. Notice that we also train our X-Adapter for Stable Diffusion v2.1 [8], which shows promising results shown as Fig. 6. For training, we select a subset of Laion-high-resolution containing 300k images for X-Adapter training. In our experiments, the input image is resized into 1024×1024 for the upgraded model and 512×512 for the base model. We utilize the AdamW optimizer with a learning rate of $1e^{-5}$ and a batch size of 8. The model is trained for 2 epochs using 4 NVIDIA A100 GPUs.

4.2. Comparisons

Experiment setting. We choose two representative plugins (ControlNet [43] and LoRA [17]), to evaluate the performance of the proposed method, since they represent two valuable applications of semantic and style control. We evaluate the performance gain our method achieves as well as plugin functionality retention. For ControlNet, we choose canny and depth to test our method under dense and sparse conditions. We utilize the COCO validation set, which contains 5,000 images, to evaluate each method. For LoRA [17], We use AnimeOutline [4] and MoXin [5] to test the style control plugin. We select 20 prompts from civitai [16] for each LoRA, generating 50 images per prompt using random seeds. To eliminate SDXL [29]'s effect on style, SDXL's prompt only focus on image's content, and X-Adapter's prompt will include LoRA's trigger words and style-related words. As for evaluation metrics, we use Frechet Inception Distance (FID) to measure the distribution distance over images generated by our method and original SDXL, which indicates image quality, as well as text-image clip scores. We also calculate the condition reconstruction score following ControlNet [43] and style similarity following StyleAdapter [38] to evaluate the plugin's functionality. The style similarity is measured between the generation of our method and the base model.

Plugin: **ControlNet**	FID ↓	CLIP-score ↑	Cond. Recon. ↑
SD 1.5 [32]	33.09	0.2426	**0.33 ± 0.16**
SDEdit [22] + SDXL	**30.86**	0.2594	0.14 ± 0.10
X-Adapter + SDXL	30.95	**0.2632**	0.27 ± 0.13
Plugin: **LoRA**	FID ↓	CLIP-score ↑	Style-Sim ↑
SD 1.5 [32]	32.46	0.25	-
SDEdit [22] + SDXL	30.11	0.2584	0.72
X-Adapter + SDXL	**29.88**	**0.2640**	**0.83**

Table 1. Quantitative evaluation against baselines.

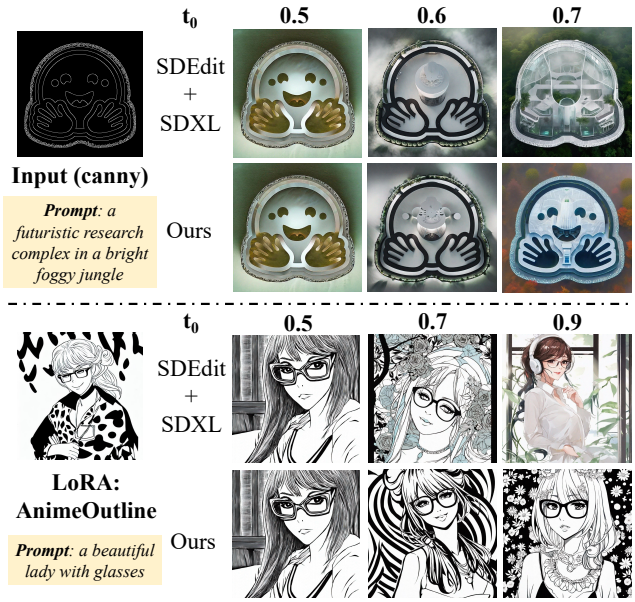


Figure 4. ***Visual Comparison to baseline under different*** t_0. We choose ControlNet [43] and LoRA [17] to evaluate different methods under semantic and style control. Specifically, we choose AnimeOutline[4], a LoRA specialized in black and white sketch generation. We sample three t_0 for each plugin. We observe that baseline loses style control (turn black and white to color) and semantic control as t_0 increases while our method maintain the controllability with the usage of X-Adapter.

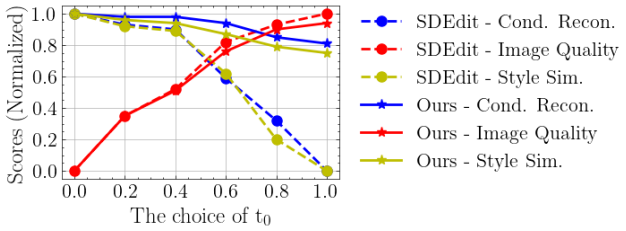


Figure 5. ***Quantitative evaluation under different*** t_0. Baseline loses style control and semantic control as $t0$ increases while our method preserves functionality of plugins

Comparison to base model. We select Stable Diffusion v1.5 [7] as our base model. The quantitative result is shown

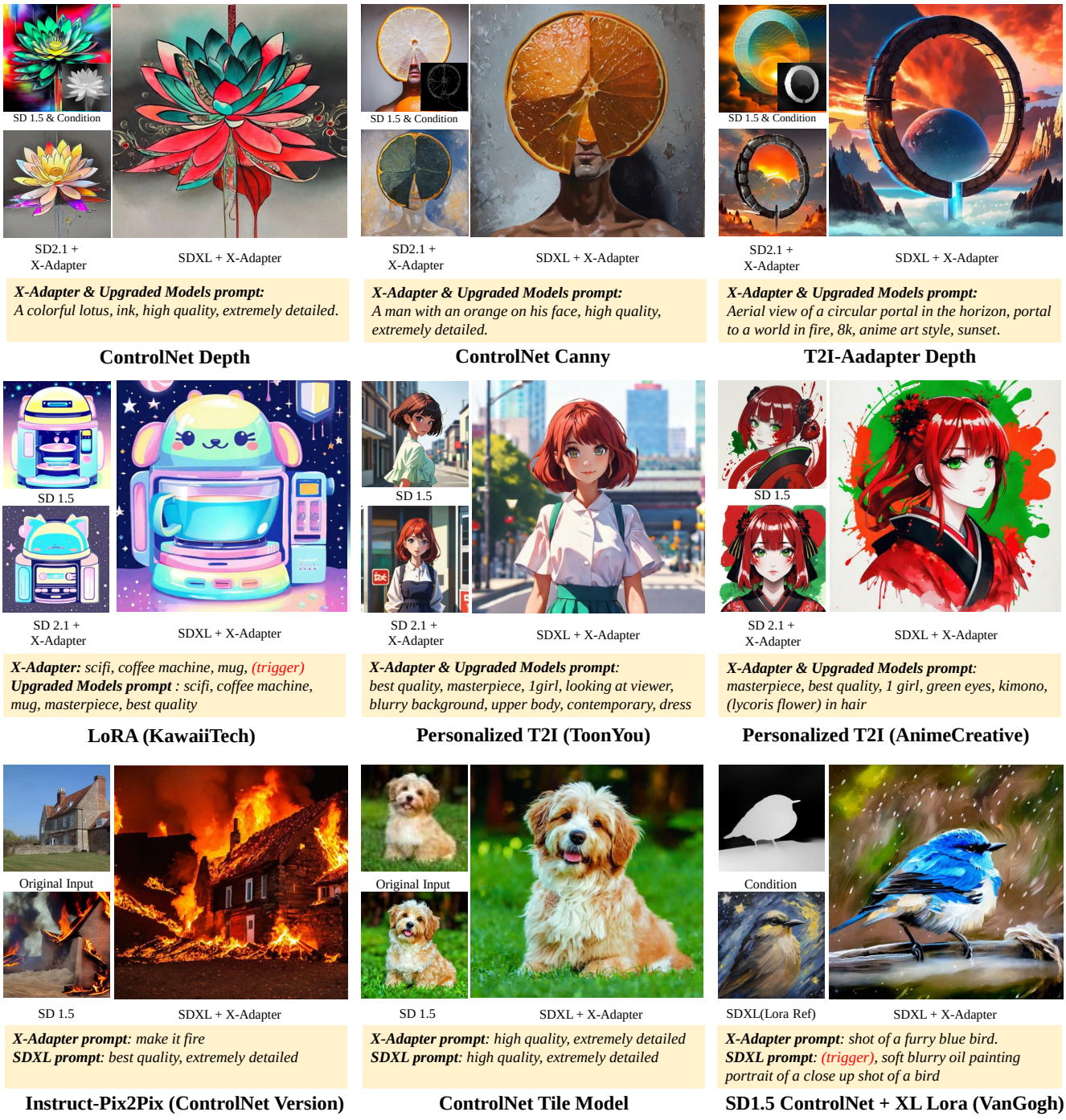


Figure 6. ***Qualitative Results on Different Plugins.*** The showcases of different results on SDXL and SD 2.1 based on the proposed X-Adapter and pre-rained SD 1.5 plugins. We show the corresponding prompts in the yellow box.

in Tab. 1. It shows that our method achieves a balance between image quality and preservation of plugin's function.

Comparison to baseline. A naive approach is to consider SDXL as an editor for the output of the base Stable Diffusion v1.5 model, similar to SDEdit [22]. We select a timestep t_0, adding noise to the base model's generation to t_0 and denoising it using the upgraded model. We evaluate it under the same experiment setting as shown in Tab.1. Note that the function of t_0 in SDEdit is similar to T_0 in our two-stage inference strategy. For both methods, the upgraded model is more influenced by the base model when t_0 is lower, obtaining more semantic features and style in-

Plugin: **ControlNet**	Result Quality ↑	Condition Fidelity ↑
SD 1.5 [32]	3.23 ± 0.12	**4.21 ± 0.32**
SDEdit [22] + SDXL	4.14 ± 0.57	2.46 ± 0.17
X-Adapter + SDXL	**4.46 ± 0.43**	3.92 ± 0.26
Plugin: **LoRA**	Result Quality ↑	Style Fidelity ↑
SD 1.5 [32]	2.93 ± 0.09	-
SDEdit [22] + SDXL	3.92 ± 0.53	3.45 ± 0.33
X-Adapter + SDXL	**4.38 ± 0.25**	**4.14 ± 0.29**

Table 2. ***User Study***. We report the user preference ranking (1 to 5 indicates worst to best) of different methods.

formation from the base model, which leads to less optimal outcomes in terms of image quality. Conversely, a higher t_0 value decreases the base model's influence, leading to improved generation quality as shown in Fig. 4. This implies that the SDE-based method loses essential semantic details and style information (*i.e.*, plugin's control) when t_0 is large, indicative of higher image quality. Conversely, X-adapter can maintain these controls and preserve the capabilities of the plugins even with a high t_0, ensuring high-quality generation with faithful plugin fidelity. To highlight the advantage of our method, we sampled six t_0 values at equal intervals between $[0, 1]$ and conducted experiments on our method and baseline under these t_0. Fig. 4 and Fig. 5 illustrate the performance of different methods. We observe that although our method shows similar visual quality compared to the baseline, it better preserves the functionality of plugins.

User study. Users evaluate the generation results of our method with ControlNet [43] and Lora [17]. For ControlNet, we collect 10 canny conditions and depth conditions, then assign each condition to 3 methods: Stable Diffusion v1.5, SDEdit + SDXL, and X-Adapter. We invite 5 users to rank these generations in terms of *"image quality"* and *"fidelity of conditions"*. For LoRA, we collect 10 prompts and also assign them to these three methods. Users rank these generations in terms of *"image quality"* and *"style similarity"*. We use the Average Human Ranking (AHR) as a preference metric where users rank each result on a scale of 1 to 5 (lower is worse). The average rankings are shown in Tab 2.

4.3. Qualitative Results on Multiple Plugins

As shown in Fig. 6, we show the qualitative results of the proposed X-Adapter on both SD 2.1 and SDXL in various pretrained plugins on Stable Diffusion v1.5 to show the advantages. We present representative examples of conditional generation (ControlNet Depth, ControlNet Canny, T2I-Adapter Depth), the personalization style (LoRA Model [2], Personalized Model [1, 6]) and the Image Editing Methods (ControlNet-based InstructPix2Pix and ControlNet Tile). Finally, we show the plugin remix in our methods, where the plugins [3] in SDXL can also directly cooperate with the Stable Diffusion v1.5 plugin (*e.g.*, ControlNet in our case).

Figure 7. ***Ablation of module to insert mapping layers***. The key to better guidance ability is to retain encoder's feature space. Prompts: *"a fantastic landscape / an apple with a lizard in it"*.

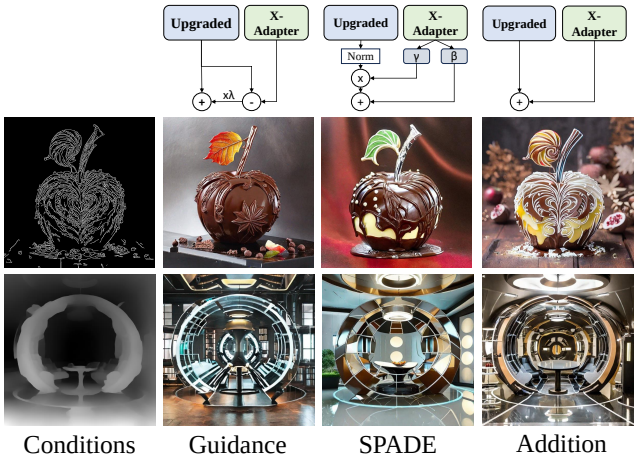


Figure 8. ***Ablation of different fusion types***. The result shows that fusing features through addition can maximize the restoration of the condition. The text prompts are: *"A chocolate apple"* and *"A research room"*.

4.4. Ablative Study

Where to insert mapping layer? We study the effect of inserting mapping layers into different modules: (1) Encoder; (2) Decoder; (3) Both encoder and decoder. Fig. 7 indicates that the decoder-only strategy shows the strongest guidance capability since it does not harm the encoder's feature space and only performs guidance during generation. See also the supplementary material for quantitative results for different module selection.

How do mapping layers guide the upgraded model? We explored three methods for integrating guidance features into the upgraded model. Given guidance feature a and upgraded model's feature b, new feature c is formed by (1) addition fusion: $c = a + b$ (2) guidance fusion: $c = b + \lambda(a - b)$ where λ can be adjusted by users (3)

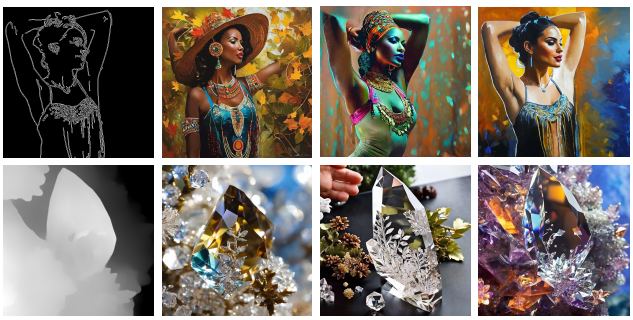


Figure 9. ***Ablation of different null probability during training***. Increasing the percentages of null text prompts in the upgraded model can enhance $X_{Adapter}$'s guidance ability. Text prompts are: *"A painting of a beautiful woman"* and *"A world of crystal"* from top to bottom.

Figure 10. ***Ablation of inference strategy.*** The result shows that X-Adapter can roughly reconstruct the condition even w/o the two-stage inference, and the two-stage inference has a better similarity. Text prompt: *"stormtrooper lecture, photorealistic"*

SPADE: $c = \gamma(a)(norm(b)) + \beta(a)$ where γ and β are two networks following SPADE [28]'s design. Fig. 8 presents a visual comparison of different ablation fusion types. We find that addition is the most effective way to provide guidance for the upgraded model.

Is using empty text important in the upgraded model? To demonstrate the effectiveness of the null-text training strategy, we train three models under 100%, 50%, and 0% null probability. Fig. 9 indicates that reducing the capability of the upgraded model during training can maximize the guidance effect of X-Adapter.

Is two-stage inference important? We study the effect of a two-stage denoising strategy by randomly sampling initial latents for X-Adatper and upgraded model. Our method still works effectively without initial latents alignment as shown in Fig. 10. Adopting two-stage sampling strategy in inference further boosts performance in terms of conditional accuracy.

4.5. Discussion

Prompt Setting. We always set clear and sufficient prompts for X-Adapter, therefore we study three different prompt settings of SDXL: (1) Sufficient prompts which are semantically consistent with X-Adapter's prompts (2) Insufficient prompts. The default insufficient prompt in this paper is "best quality, extremely detailed". (3) Conflicting prompts which change the meaning of X-Adapter's prompts. Fig. 11 shows that our method can still maintain overall layout and style consistency even in case of prompt conflict.

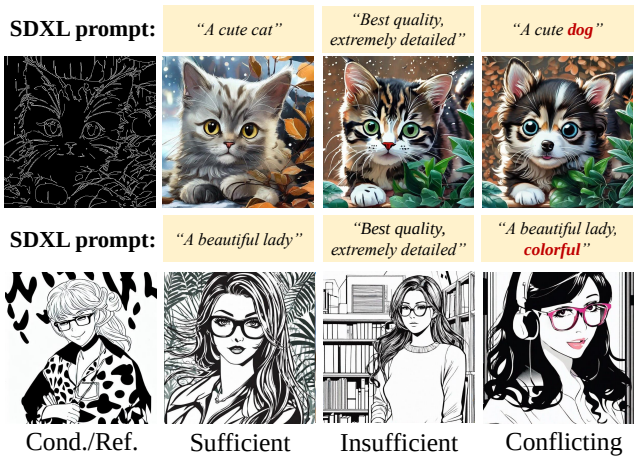


Figure 11. ***Prompt setting***. Our method can still ensure the overall layout and style consistency even in case of prompt conflict. LoRA[17] used here isAnimeOutline[4], an expert in black and white sketch generation. X-Adapter's text prompts are: *"A cute cat"* and *"A beautiful lady, (trigger words)"* from top to bottom.

Plugin Remix. Our method naturally supports plugins from both X-Adapter (*e.g.* SD1.5 [7]) and upgraded model (*e.g.* SDXL [29]) since we retain all connectors by freezing parameters of these two models. The bottom right picture of Fig. 6 shows a combination of Stable Diffusion v1.5's ControlNet and SDXL's LoRA, generating results that follow LoRA's style and condition's semantics. It indicates that our method can bridge community resources across different diffusion model versions (*e.g.* SD1.5, SD2.1 [8], SDXL).

Limitation. Although our method achieves impressive results, it still has some limitations. For some plugins to generate personalized concepts, *e.g.*, IP-Adapter [39], our method might not maintain the identity well. We give examples in the supplementary material for visualization. This is because the custom plugins work on the text-encoder other than the feature space concepts that are not directly injected into the upgraded model other than fused as guidance. Since our method has already made some universal plugin upgrades, we leave the capability of the concept customization as future work.

5. Conclusion

In this paper, we target a new task of upgrading all the downstream plugins trained on old diffusion model to the upgraded ones. To this end, we propose X-Adapter, which comprises a copied network structure and weights of the base model, and a series of mapping layers between two

decoders for feature mapping. During training, we freeze the upgraded model and set its text prompt to empty text to maximize the function of X-Adapter. In testing, we propose a two-stage inference strategy to further enhance performance. We conduct comprehensive experiments to demonstrate the advantages of the proposed methods in terms of compatibility and visual quality.

References

[1] Animecreative. https://civitai.com/models/146785. 7

[2] Kawaiitech. https://civitai.com/models/94663. 7

[3] Vangoghportraiture. https://civitai.com/models/157794. 7

[4] Animeoutline. https://civitai.com/models/16014. 5, 8

[5] Moxin. https://civitai.com/models/12597. 5

[6] Toonyou. https://civitai.com/models/30240. 7

[7] Stability AI. https://huggingface.co/runwayml/stable-diffusion-v1-5, . 2, 5, 8

[8] Stability AI. https://huggingface.co/stabilityai/stable-diffusion-2-1-base, . 2, 5, 8

[9] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arxiv:2310.19512*, 2023. 2

[10] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv preprint arxiv:2105.05233*, 2021. 2

[11] 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. 3

[12] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 2, 3, 4

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016. 4

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[15] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. pages 2790–2799, 2019. 3

[16] https://civitai.com/. civitai. https://civitai.com/, 2013. 5

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 3, 5, 7, 8, 1

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arxiv:1312.6114*, 2013. 2, 3

[19] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arxiv:2107.00630*, 2021. 2

[20] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. *arXiv preprint arxiv:2301.07093*, 2023. 2

[21] Shanyuan Liu, Dawei Leng, and Yuhui Yin. Bridge diffusion model: bridge non-english language-native text-to-image diffusion model with english communities. *arXiv preprint arXiv:2309.00952*, 2023. 4

[22] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arxiv:2108.01073*, 2022. 2, 4, 5, 6, 7

[23] Midjourney. https://www.midjourney.com/. 2

[24] Chong Mou, Xintao Wang, Liangbin Xie, 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 arXiv:2302.08453*, 2023. 2, 3, 4, 1

[25] 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. *arXiv preprint arxiv:2112.10741*, 2022. 2

[26] OpenAI. Dall-e2. https://openai.com/dall-e-2, . 2

[27] OpenAI. Dall-e3. https://openai.com/dall-e-3, . 2

[28] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. *arXiv preprint arxiv:1903.07291*, 2019. 8

[29] 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. 2, 3, 4, 5, 8

[30] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv preprint arxiv:2303.09535*, 2023. 2

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. 2021. 2, 3, 5, 7

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *arXiv preprint arxiv:1505.04597*, 2015. 3

[34] 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 arxiv:2208.12242*, 2023. 2, 3

[35] 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 arxiv:2205.11487*, 2022. 2

[36] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arxiv:1503.03585*, 2015. 2

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arxiv:1706.03762*, 2017. 2, 3

[38] Zhouxia Wang, Xintao Wang, Liangbin Xie, Zhongang Qi, Ying Shan, Wenping Wang, and Ping Luo. Styleadapter: A single-pass lora-free model for stylized image generation. *arXiv preprint arxiv:2309.01770*, 2023. 5

[39] 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 preprint arxiv:2308.06721*, 2023. 2, 3, 8, 1

[40] Ge Yuan, Xiaodong Cun, Yong Zhang, Maomao Li, Chenyang Qi, Xintao Wang, Ying Shan, and Huicheng Zheng. Inserting anybody in diffusion models via celeb basis. *arXiv preprint arXiv:2306.00926*, 2023. 3

[41] Binjie Zhang, Yixiao Ge, Xuyuan Xu, Ying Shan, and Mike Zheng Shou. Taca: Upgrading your visual foundation model with task-agnostic compatible adapter. *arXiv preprint arxiv:2306.12642*, 2023. 3

[42] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arxiv:2309.15818*, 2023. 2

[43] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. 2023. 2, 3, 4, 5, 7, 1

[44] Hengyuan Zhao, Hao Luo, Yuyang Zhao, Pichao Wang, Fan Wang, and Mike Zheng Shou. Revisit parameter-efficient transfer learning: A two-stage paradigm. *arXiv preprint arxiv:2303.07910*, 2023. 3

X-Adapter: Adding Universal Compatibility of Plugins for Upgraded Diffusion Model

Supplementary Material

A. Ablation on module selection

We provide quantitative results of ablation on module to insert mapping layers. Tab. 1 indicates that the decoder-only strategy shows the strongest guidance capability since it does not harm the encoder's feature space and only performs guidance during generation.

Image prompt IP-Adapter Ours

Figure 1. ***Limitations***. In IP-Adapter [39], although our method can produce relatively identity-consistent results, the details, *e.g.*, clothes, are still different from the original model.

B. Limitation

For plugins to generate personalized concepts, *e.g.*, IP-Adapter [39], our method might not maintain the identity well like shown in Fig. 1. This is because the custom plugins work on the text-encoder other than the feature space concepts that are not directly injected into the upgraded model other than fused as guidance.

C. Qualitative result

We provide more qualitative result on Controlnet [43], T2I-Adapter [24] and LoRA [17] as shown in Fig. 2

Plugin: **ControlNet**	FID ↓	CLIP-score ↑	Cond. Recon. ↑
Encoder & Decoder	38.37	0.26	0.24 ± 0.15
Encoder only	37.32	0.26	0.23 ± 0.14
Decoder only	**30.95**	**0.26**	**0.27 ± 0.13**
Plugin: **LoRA**	FID ↓	CLIP-score ↑	Style-Sim ↑
Encoder & Decoder	36.71	0.26	0.79
Encoder only	35.54	0.26	0.80
Decoder only	**29.88**	**0.26**	**0.83**

Table 1. ***Ablation on module to insert mapping layers.*** For encoder only and decoder only, we use the same mapping layers to map the frozen base model to the upgraded model's encoder and decoder separately. For both the encoder and decoder, we use two identical mapping layers and insert them into the encoder and decoder.

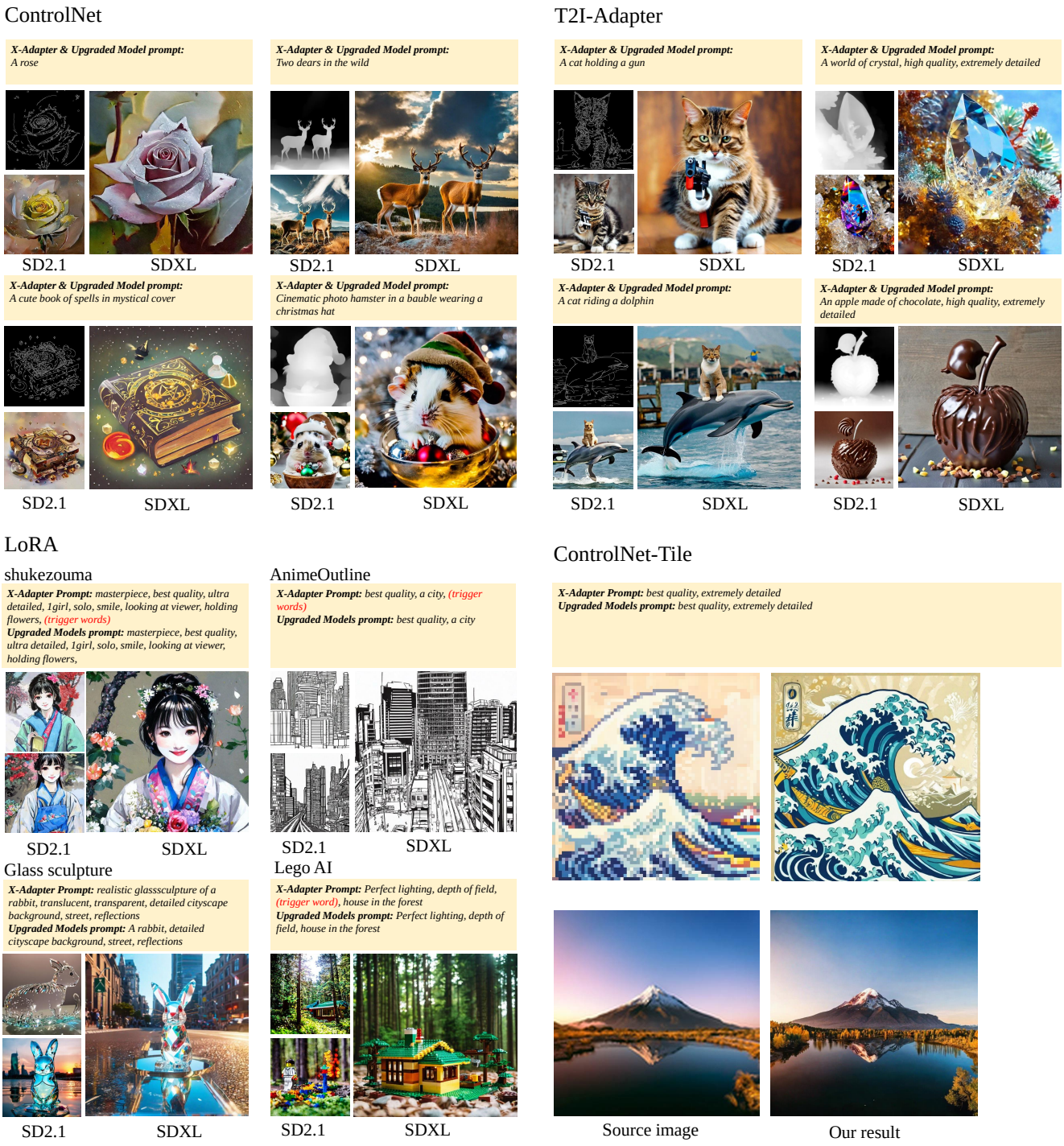


Figure 2. ***Qualitative Results on Different Plugins.***