Versatile Diffusion: Text, Images and Variations All in One Diffusion Model – Supplementary –

Xingqian Xu[1], Zhangyang Wang[2,3], Eric Zhang[1], Kai Wang[1], Humphrey Shi[1,3]
[1]SHI Labs @ Georgia Tech & UIUC & Oregon, [2]UT Austin, [3]Picsart AI Research (PAIR)
`https://github.com/SHI-Labs/Versatile-Diffusion`

1. Application Details

1.1. Disentanglement of Style and Semantic

The disentanglement application conducts controllable image-variation, supported by the image-variation flow of VD. Such flow consists of AutoKL, CLIP image encoder, and VD's diffuser with image data layers and image context layers. The core strategy of the disentanglement is to manipulate the 257×768 CLIP image context embedding, which guided the diffusion process via cross-attention. Recall that these embeddings are generated by the visual transformer [3], which begins with one global feature vector followed by 256 local feature vectors corresponding to image local patches. We first split the vector into the single global vector and the following 256 local vectors. We keep the global vector untouched and compute the principal components from the rest of the feature vectors. When manipulating the context embeddings, we notice that the first couple of principal components (*i.e.* the major principal components of the matrix) hold the style information (*i.e.* color, art, stroke styles), and the remaining principal components hold the semantic information (*i.e.*, objects, object locations, identity). Thus, in practice, we generated image variations with style focuses from the guidance of the low-rank context embedding that hosts only major principal components. And we generated image variations with semantic focuses when we removed these major principal components from context embeddings. In Figure 1, we show additional qualitative results, in which we standardize the disentanglement with five levels:

(a) 0 represents normal image-variation.
(b) -1 and -2 are semantic-focused by removing one and two major principal components.
(c) 1 and 2 are style focuses results corresponding to keeping only 10 and 2 major principal component.

In practice, we also notice that principal components after order 50 have little effect on results. Therefore, we can speed up the disentanglement PCA by just computing the first 50 principal components and then conducting the manipulation. For the global feature vector, we notice that it mainly serves as a semantic feature that controls object location information. Hence, removing it may negatively impact image structure (see Figure 2) but may be useful in some art generation cases. We encourage researchers to explore further the low-rank subspace of the CLIP Image embedding for more exciting applications.

1.2. Dual-context Blender

The dual-context blender for VD is to generate images through the guidance of one image and one text prompt. Theoretically, such an application can also be used to create new text/sentences, but the results are less exciting than creating new images. As mentioned in the main article, the dual-context blender, or the multi-context blender, can be achieved by ensembling two models in which we mix the diffusion steps from one model after another (type A), or weighted sum up both models' outputs (type B). However, in practice, we notice that such approaches may cause structure distortions and highlight wrong semantics despite doubling the model usage. Unlike these simple ensembling methods, VD can carry out this task via a much deeper level of mixing due to its multi-flow multimodal framework. As mentioned in the main article Section 3.2, our framework has three layer groups: global, data, and context. When generating images, features diffuse through the shared data layers (*i.e.* ResBlocks), and then mix up via different context layers with two options: layer-level mixing or attention-level mixing. In layer-level mixing, we diffuse features through different context layers (*i.e.* cross-attention) that follow a preset schedule. For example, we diffuse features through one image cross-attention, then a text cross-attention, *etc.*

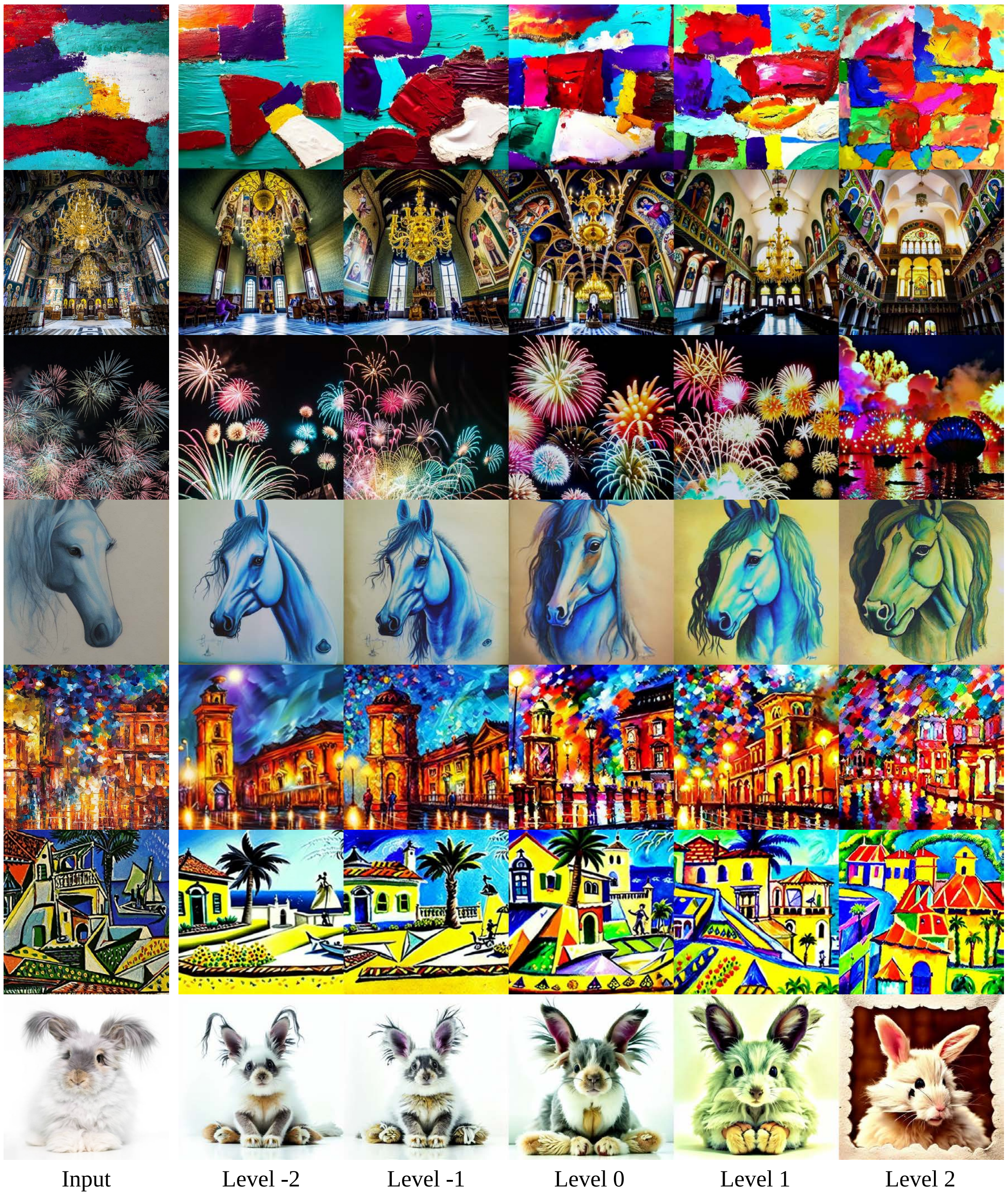


Figure 1: Additional figures that show the performance of our proposed disentanglement application with different levels.

In attention-level mixing, both immediate features after context layers are included via a weighted sum and then passed to the network's next block (See Figure 3).

We believe that the success of the dual-context blender heavily relies on VD's multi-flow design that can merge contexts in a harmonized way. An example shows in Figure 4 in which we generate an image using *a car* as image context and a prompt *a double-decker bus* as text context. Such a case manually brings challenges in mixing, since a Benz car in the image and a double-decker bus described in the prompt have completely different shapes. However, attention-level mixing nicely resolves such conflict and we notice a smooth transition between these two contexts with increasing mixing rates. On the

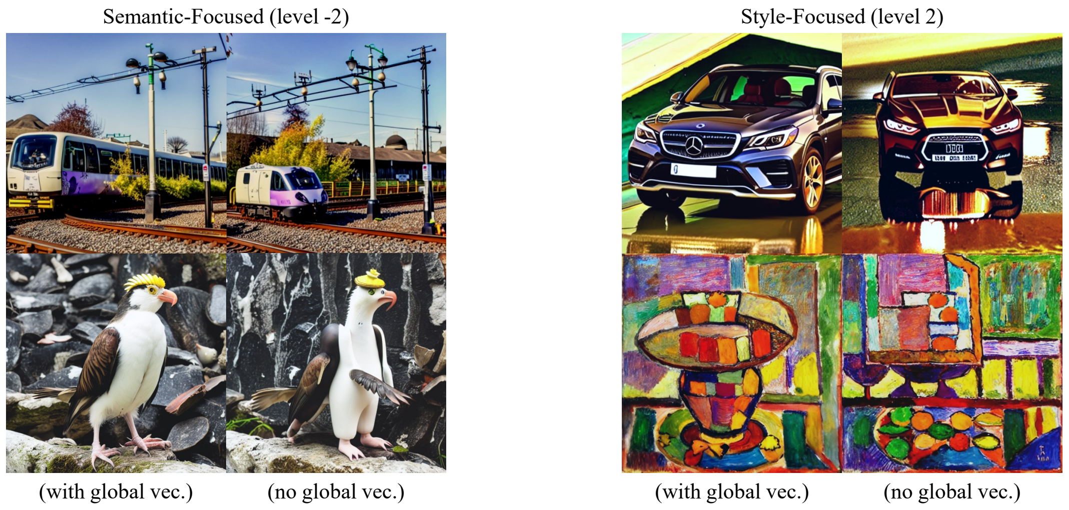


Figure 2: This figure shows four comparisons between samples generated via context with the global vector and without the global vector. The two left cases are semantic-focused outputs, and the two right cases are style-focused outputs.

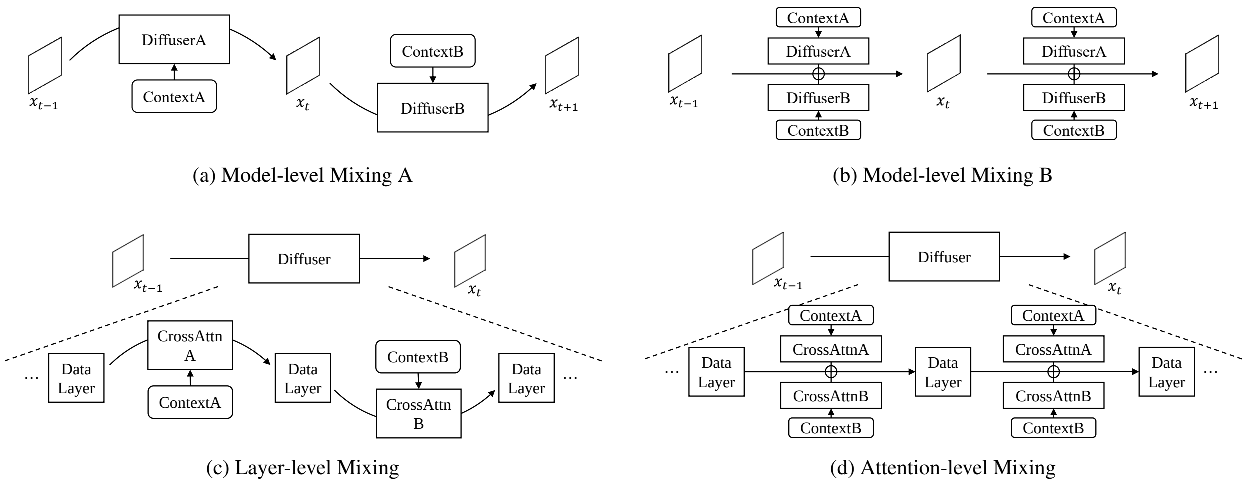


Figure 3: The graphic explanation of the three mixing strategies we mentioned in our dual-context blender. Both (a) and (b) are two types of model-level mixing.

other hand, our results indicate that layer-level mixing slightly underperforms attention-level mixing as the generated vehicle shows some noticeable distortion (see the wheels in the second line). Lastly, we show the results from model-level mixing using two of our baseline models SDv1.4 and SD-variation, which perform the worst among all three methods. Given these results, we conclude that VD is critical for the success of our dual-context blender tasks, in which its multi-flow multimodal network framework is the key to effectively resolving potential conflict and merging various contexts.

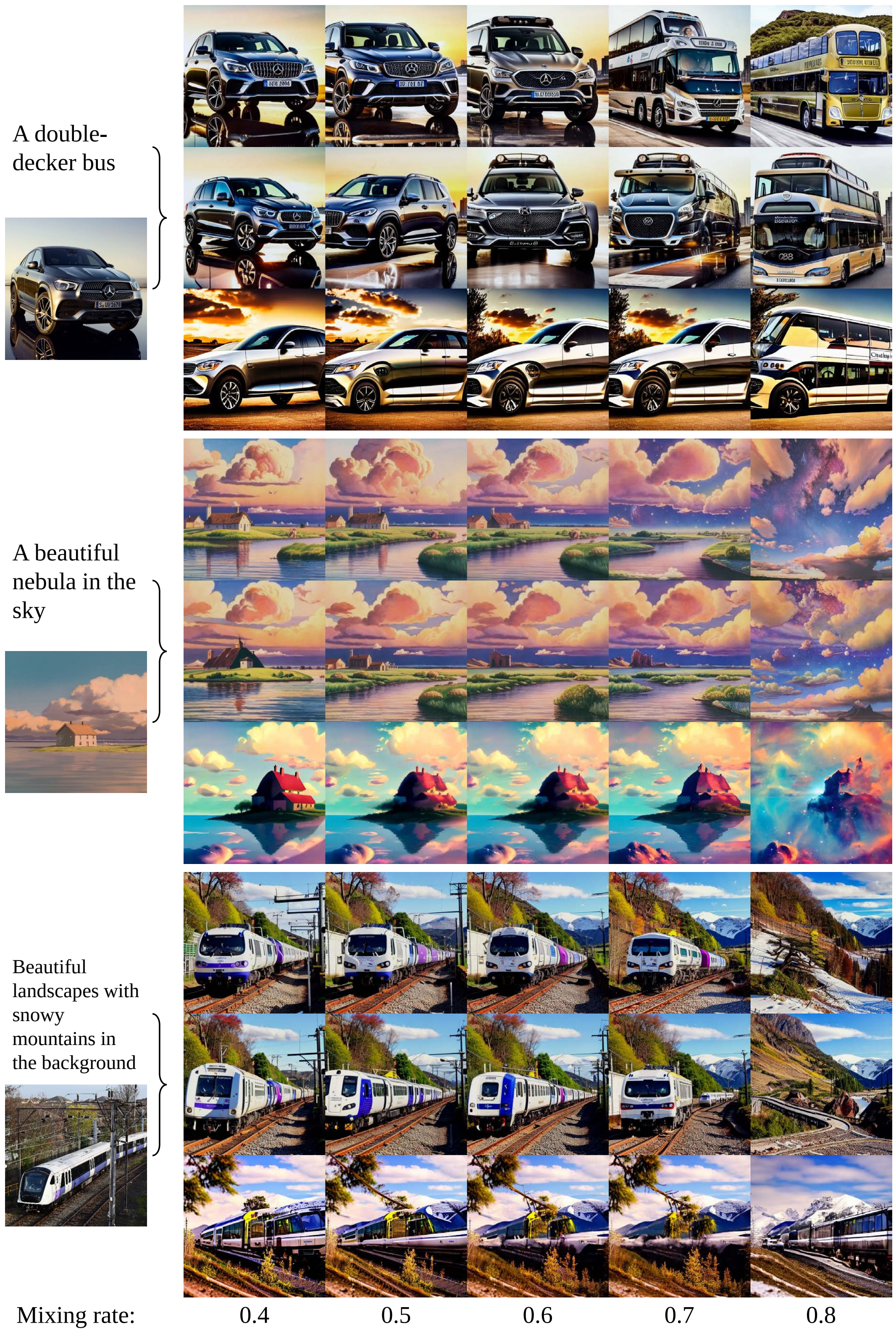
1.3. Multi-context Blender with Optional Masks

Multi-context blender is an extension of our dual-context blender with the following changes:

(a) It takes more than one image as a concatenated context of image type.

(b) It accepts additional scale control on each of the input images.

(c) It allows adding individual masks to precisely control the generated output based on reference images.



A double-decker bus
A beautiful nebula in the sky
Beautiful landscapes with snowy mountains in the background
Mixing rate:
0.4
0.5
0.6
0.7
0.8

Figure 4: Additional figures that show the performance of our proposed dual-context blender. The horizontal axis shows the mixing rate we use in these samples: small mixing rates lead to image-focused (to the left), and large mixing rates lead to text-focused (to the right). For each sample, we show three rows, from top to bottom are the results of attention-level mixing, layer-level mixing, and model-level mixing, in which the top row (attention-level mixing) is the best.

Adding an extra image as reference context is actually simpler than adding an extra context type. Specifically for VD, context from multiple images can be concatenated, forming a more extended sequence of context embedding that later serves as input to content layers. For example, context embedding one image is a 257×768 embedding in which 257 is the number of embedding vectors and 768 is the embedding channel. Context embedding for two images is simply a concatenated feature of two images embedding along the number dimension, making it 514×768, and so-on-so-forth for more images. Therefore, such operations do not alter any mixing strategies that we used in the dual-context blender.

To make more precise control of images used in our multi-context blender, we involved two controllable parameters: image scales and image masks. Image scales are simple multipliers associated with underlining image context embeddings, while image masks involve more complex designs. We notice that a naive solution to replace contents in masks with zeros may confuse the model of generating images with black patches. Thus we altered the CLIP network, in which the raw image features after the first convolution projection of ViT [3] and the input positional encodings are filled with zeros according to masks before inputting the transformers. As a result, we successfully involved scales and masks in our blender application. More results can be found in Figure 5.

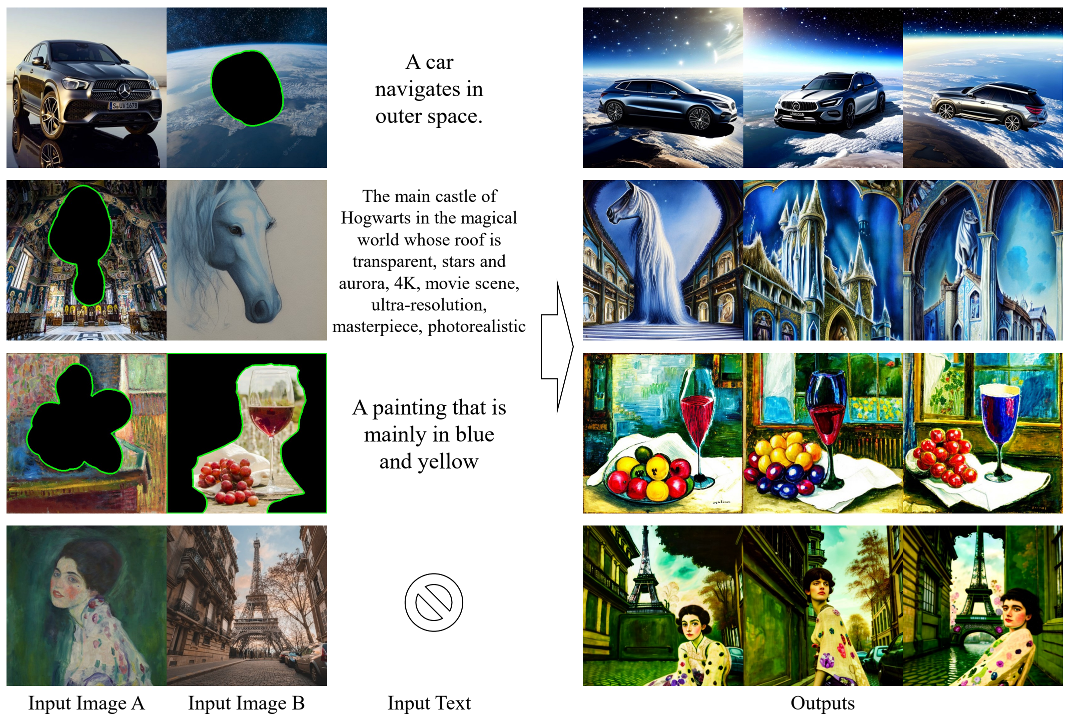

A car navigates in outer space.
The main castle of Hogwarts in the magical world whose roof is transparent, stars and aurora, 4K, movie scene, ultra-resolution, masterpiece, photorealistic
A painting that is mainly in blue and yellow
Input Image A
Input Image B
Input Text
Outputs

Figure 5: Additional examples generated by VD's multi-context blenders. As mentioned in the paper, our multi-context blender uses multiple images with optional masks adding an extra text prompt (also optional) to guide its generation process. Its outputs are then a blender of all input context.

1.4. Editable I2T2I

Since VD supports both image-to-text and text-to-image, one heuristic image editing approach we can do is to edit images with the following steps: **(a)** *convert image to text*, **(b)** *edit text*, and **(c)** *convert text back to the image*. We named this approach *image-to-text-to-image* **(I2T2I)**. Although the principle of I2T2I is simple, we notice that in practice, many issues may negatively affect the outputs, making them less robust than expected.

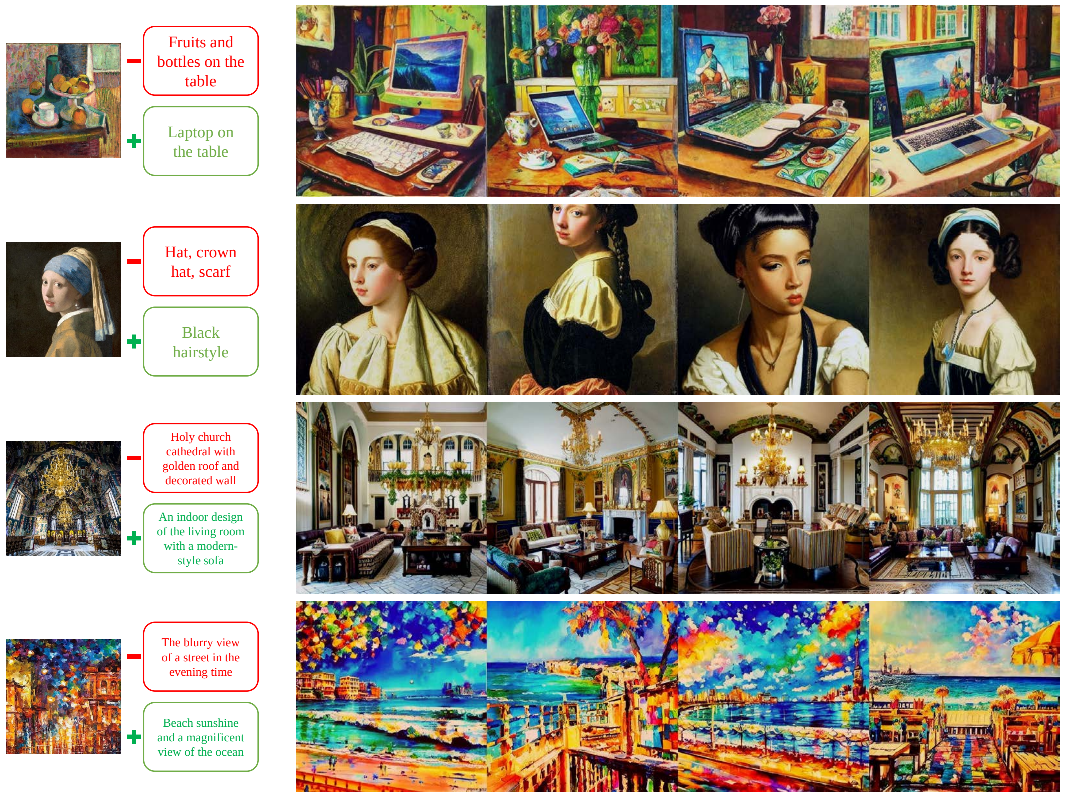
Unlike inpainting or multi-context blending, I2T2I requires no object masks because one design goal of I2T2I is to let it automatically locate and substitute objects following the prompt instruction. Meanwhile, I2T2I's output images do not match its input images pixel by pixel, which is a result of semantic distillation and content creation. Figure 6 shows the prototype results of our I2T2I in which old contents inside the image are removed and re-placed via prompt editing. To the best of our knowledge, this is the first attempt at creating and editing images by combining image-to-text, text editing, and then text-to-image. Yet we notice the following issues that may dramatically decrease the performance:

(a) The output quality is affected by both image-to-text and text-to-image performance. The failure of either one of the sub-procedures will result in unsatisfactory results.

(b) Sometimes, direct editing of the generated text could be infeasible, as there is no guarantee that the text contains the descriptions we would like to modify.

(c) Image-to-text is a process of information distillation, while text-to-image is a process of information creation. Although such properties bring great flexibility in I2T2I editing, they may differ from general users' demands because they may like to keep more content from the reference images.

To overcome these issues, we tackle these issues with the following solution. Instead of editing the text directly, we actually modify the latent text vectors as a solution to b) issue. Speaking with details, in our editing experiment, we prepare a negative and positive prompt to do the editing. The negative prompt describes image content that needs to be removed, and the positive prompt describes the content to add. When the text latent vector is ready (using image-to-text), before converting it into text, we project it on the normal space of the negative prompt latent vector and sum it up with the positive prompt latent vector. To further strengthen the positive prompt, we also compute its CLIP embedding and concatenate it with the modified prompt embedding, and then guide the image generation. Meanwhile, we adopt the ideas from our disentanglement and dual-context blender, in which we compute the style disentangled image context and use it as secondary guidance in the generation process with a 0.66 mixing rate. The final performance of I2T2I can be found in Figure 6.



−
Fruits and bottles on the table
+
Laptop on the table
−
Hat, crown hat, scarf
+
Black hairstyle
−
Holy church cathedral with golden roof and decorated wall
+
An indoor design of the living room with a modern-style sofa
−
The blurry view of a street in the evening time
+
Beach sunshine and a magnificent view of the ocean

Figure 6: This figure shows the performance of our proposed image editing method I2T2I with negative and positive prompts. (see Section 1.4)

2. Image-Variation Analysis and Beyond

2.1. Unconditional Guidance

We also did an in-depth investigation on the influence of the unconditional guidance (*i.e.* classifier-free guidance) of the image-variation. Recall that such mechanism is first involved in class-guided generation [5, 11] and later largely adopted by text-to-image models such as [11, 10, 12]. Here we recap the core math in Equation 1:

$$y = y_u + (y_c - y_u) * s, \quad y_u = G(c_{\text{uncond}}), \quad y_c = G(c_{\text{cond}}) \tag{1}$$

in which y is the final output, s is the unconditional guidance scale, y_u is the unconditional output from generator G using unconditional context c_{uncond}, and y_c it the conditional output from G and c_{cond}. For text-to-image, c_{uncond} are usually set to the text embeddings encoded from empty strings. For image-variation, we have two options:

(a) CLIP embeddings of empty images with all zeros

(b) All-zero embeddings

As shown in Figure 7, both methods have pros and cons: Option (a) tends to highlight content and style in the reference image, and thus its results are more art-focused with better color contrast. Option (a) may also yield slightly better-performing disentanglement and dual-context because it sensitively captures details from input contexts and "magnify" them in the output. While this option sometimes may "over-react", which results in unbearably color and structure distortions. Conversely, Option (b) is a more robust solution that performs better on photorealistic inputs and generates outputs closer to reference images with fewer distortions.

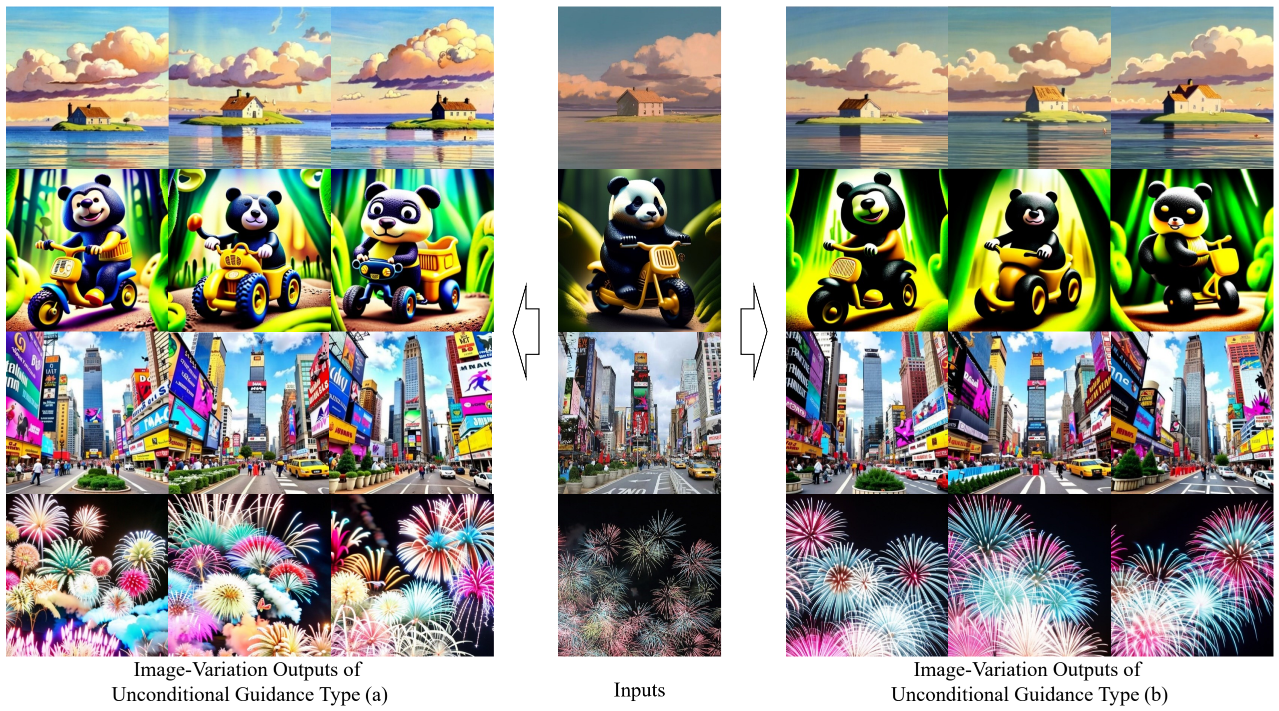


Image-Variation Outputs of Unconditional Guidance Type (a)
Inputs
Image-Variation Outputs of Unconditional Guidance Type (b)

Figure 7: This figure shows the image-variation performance of two types of unconditional guidance described in Supplementary Section 2.1. The top two rows are cases where type (a) yields better results, while the bottom two rows show that type (b) yields better results.

2.2. Image-Variation with ControlNet

ControlNet [15] and similar techniques such as [9, 6] have recently proposed a practical image editing solution involving pretrained text-to-images models and adaptive networks. One benefit of ControlNet is once an adaptive network (*i.e.* control

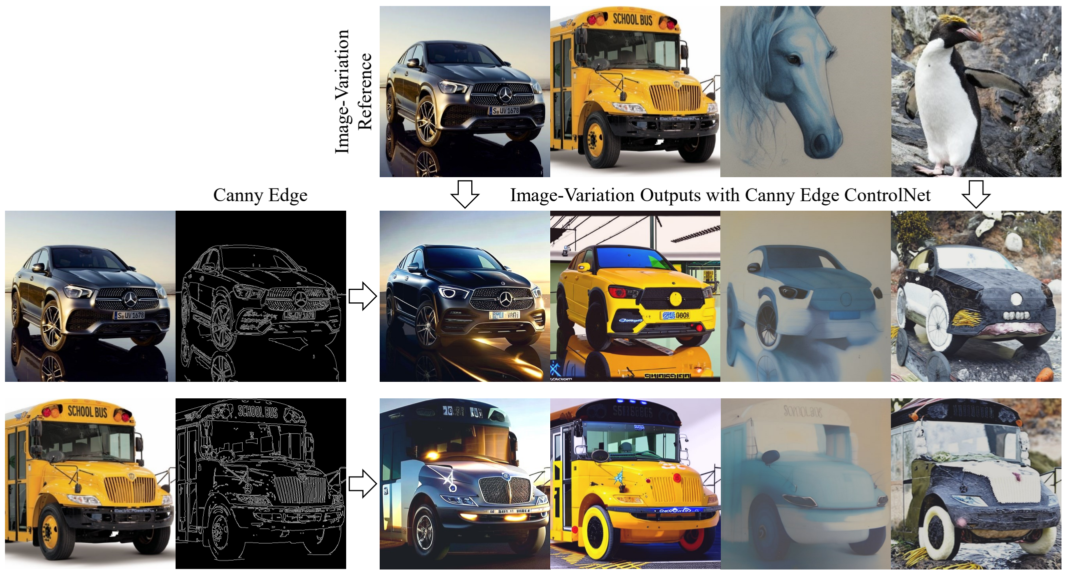


Image-Variation Reference
Canny Edge
Image-Variation Outputs with Canny Edge ControlNet
SCHOOL BUS

(a) VD's Image-variation demo with canny edge ControlNet

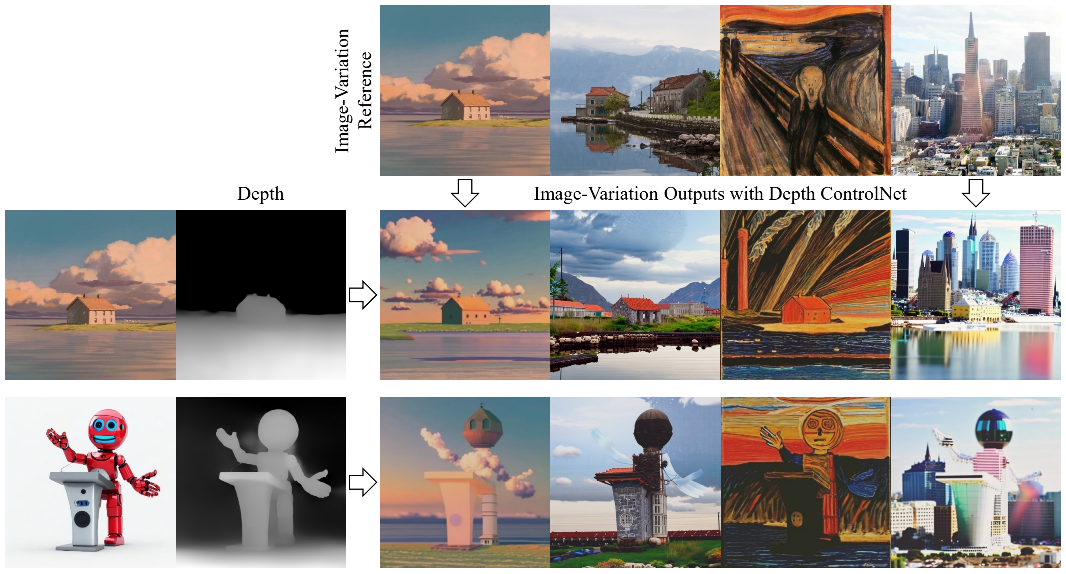


Image-Variation Reference
Depth
Image-Variation Outputs with Depth ControlNet

(b) VD's Image-variation demo with depth ControlNet

Figure 8: This figure shows the demo of our VD's image-variation results with ControlNet. A new type of image CLIP, CLIP-PA, is used to generate these results.

network) is set up, it can be easily transferred to other text-to-image models without further effort in training. Such convenience inspires us to combine the same adaptive network strategy with VD's image-variation, forming a new application for prompt-free controllable image generation.

In Figure 8, we show the performance of this new application with canny edge and depth ControlNet. The usage of these ControlNet closely follows text-to-image approaches:

(a) We first prepare a well-trained image variation model under VD's framework.
(b) We download the pretrained ControlNets and load them together with VD.
(c) We then control the image-variation process under the guidance of these ControlNets just like in text-to-image. The sole difference is we don't need any prompts.

One thing to notice is that the image-variation model we use for ControlNet slightly alters from the default version of VD, in which we remove the positional embedding layers from the CLIP image encoder to make it a position-agnostic CLIP (CLIP-PA). We then prune VD, letting the image-variation flow, and finetune it with the new CLIP-PA. The finetuned checkpoint is then a position-agnostic image-varation model. Once we have this new model, we can add ControlNet's adaptive network as text-to-image approach. The final outputs will then be a combination of ControlNet's structure hint and VD's semantic and style.

3. Data and Training

3.1. Laion2B Prompt Cleaning

As mentioned in the main article Section 4.1, we cleaned text prompts from Laion2B in order to include Optimus VAE for the to-text flows in VD. Our rules of cleaning the prompts are the followings:

(a) Remove all HTTP links, URLs, and email addresses

(b) Remove HTML syntax.

(c) Remove unnecessary contents included by square or curly brackets.

(d) Remove unnecessary symbols such as dashes, slashes, and underscores.

(e) Remove all kinds of quotes but keep *'s*.

By cleaning these captions, we were able to train VD's to-text flows in a more robust way. We did not apply such prompt cleaning when training VD on its to-image flows.

3.2. Alternative Training

In Section 4.2, we mentioned that VD's training follows a progressive rule in which we train a single-flow VD, a dual-flow VD, and finally, the four-flow VD in order. Recall that VD's single-flow model is an image-variation model, which means that image-variation is the "beginning task" VD first learns. This happened to be the rule we followed in the main paper, but it did not stop other possible ways of training. In fact, training VD can be more flexible. To demonstrate such flexibility, we alternatively trained a VD in which we set text-to-image as the "beginning task". We compare the final results of this alternative VD model (labeled as VD-Alt) with the paper model (labeled as VD) in Figure 10, in which both yield to similar performance.

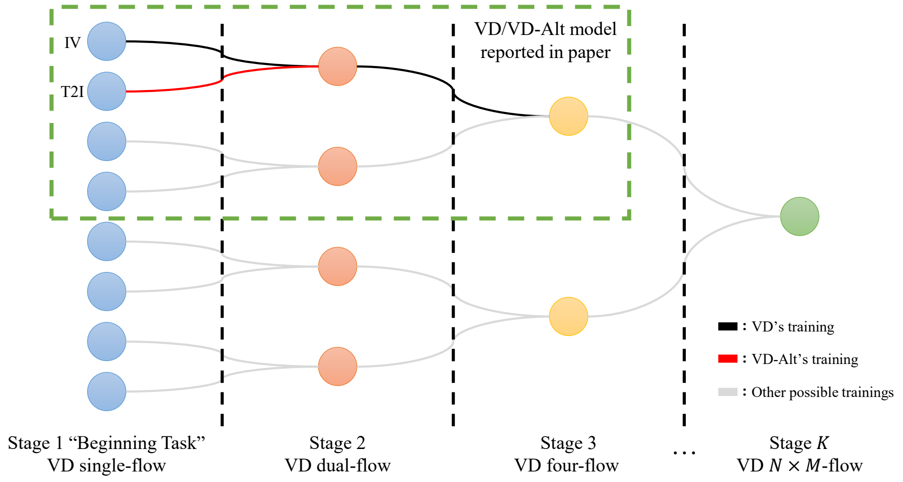

IV
T2I
VD/VD-Alt model reported in paper
: VD's training
: VD-Alt's training
: Other possible trainings
Stage 1 "Beginning Task" VD single-flow
Stage 2 VD dual-flow
Stage 3 VD four-flow
...
Stage K VD $N \times M$-flow

Figure 9: A graphic explanation of possible ways of training the generalized VD with M types of context that supports N types of outputs. The green dash box crop out the current VD/VD-Alt mentioned in the main paper and the supplementary. VD's training shows the black paths starting from IV (image-variation), while VD-Alt's training shows an altered red path from T2I (text-to-image).

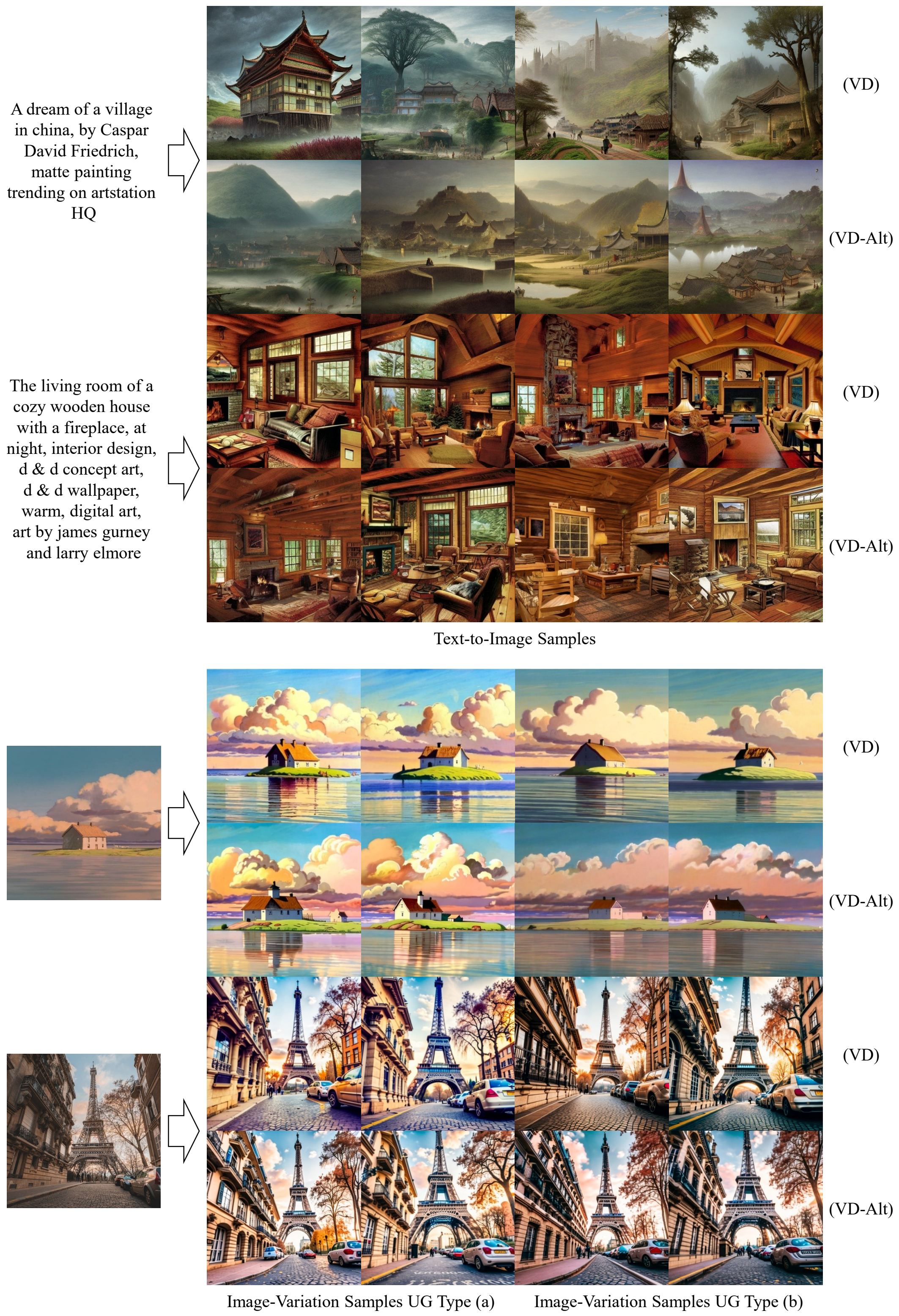

A dream of a village in china, by Caspar David Friedrich, matte painting trending on artstation HQ
(VD)
(VD-Alt)
The living room of a cozy wooden house with a fireplace, at night, interior design, d & d concept art, d & d wallpaper, warm, digital art, art by james gurney and larry elmore
(VD)
(VD-Alt)
Text-to-Image Samples
(VD)
(VD-Alt)
(VD)
(VD-Alt)
Image-Variation Samples UG Type (a)
Image-Variation Samples UG Type (b)

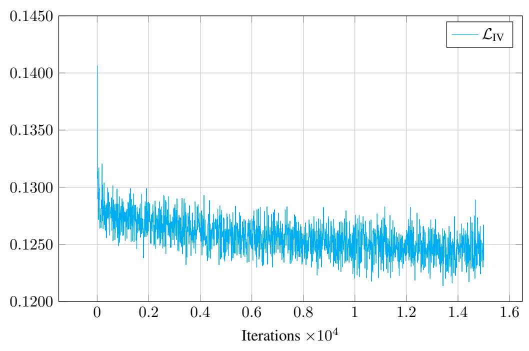
Figure 10: The qualitative comparison between VD and VD-Alt on text-to-image and image-variation. For image-variation, We show both results using unconditional guidance (UG) type (a) and (b) described in Supplementary Section 2.1.

The success of VD-Alt reveals that there may exist many feasible training rules for the multi-flow multimodel diffusion models with M context N outputs. A graphic explain of the collections of rules is illustrated in Figure 9. Given that our VD and VD-Alt all yield good performance, we prompt researchers to further explore the flexibility of training VD, which could also be one of the exciting properties of universal AI.

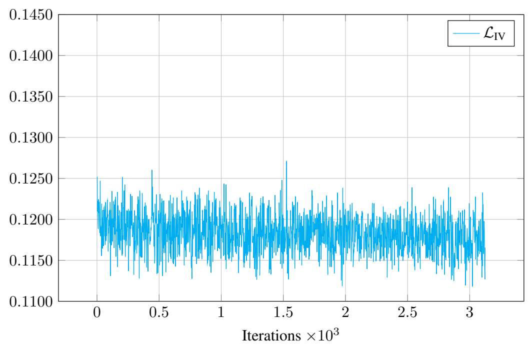
3.3. Loss Curves

We reveal VD's train-time loss curves in Figure 11. All experiments were carried out using a single node with 8 A100 GPUs (80G memory). To make the effective batch size matches the batch size we mentioned in the main paper Section 4.2 (*i.e.* 2048, 1024, and 512), we utilized the gradient accumulation technique in which we performed multiple backpropagations with one gradient update. The batch per GPU for single backpropagations are 64 for resolution 256 and 16 for resolution 512. The gradient accumulation loop can then be calculated as:

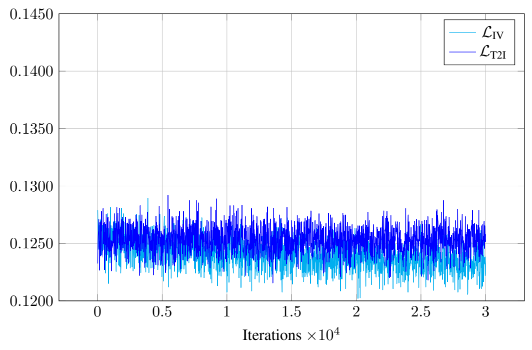
$$\text{Gradient Accumulation Loop} = \frac{\text{Effective Batch Size}}{\text{Batch per GPU} \times 8} \quad e.g. = 4 \text{ for single-flow 256 training} \tag{2}$$



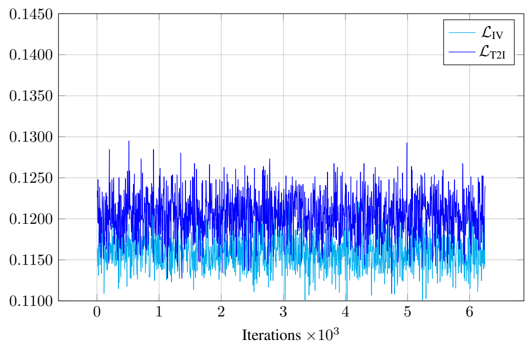
(a) VD single-flow trained on Laion2B Resolution 256



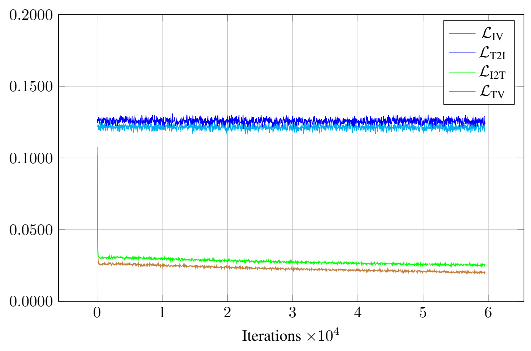
(b) VD single-flow trained on Laion2B Resolution 512



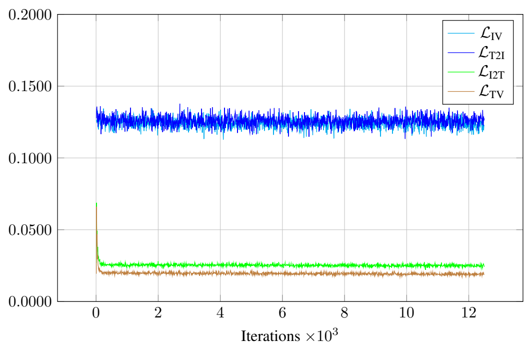
(c) VD dual-flow trained on Laion2B Resolution 256



(d) VD dual-flow trained on Laion2B Resolution 512



(e) VD four-flow trained on Laion2B Resolution 256



(f) VD four-flow trained on COYO Resolution 512

Figure 11: This figure shows the loss curves of VD single-flow, dual-flow, and four-flow during their training phases.

4. Text-Variation Performance

We show the performance of our text-variation results in Table 1, in which a total number of 8 outputs are sampled from VD for each input.

Inputs	Outputs			
blue and yellow balloons in the sky	two blue balloons and yellow balloons flying in the blue sky and the yellow sky in the sky above the blue sky.	two blue balloons and blue balloons flying under the yellow star and the blue and the yellow lights in the sky.	blue balloons and the yellow blue balloons and blue sky blue sky behind the sky.	two balloons blue and the yellow balloons and blue balloons blue up the yellow sky three above the sky in the sky.
	yellow blue balloons and two blue balloons and the yellow balloons in the sky above blue blue.	two blue balloons and the blue balloons in the blue and yellow balloons in the yellow and the green sky.	two blue balloons and yellow balloons yellow in the blue sky and the large sky at the sky.	two blue balloons and blue lights in the yellow and blue sky blue sky near the sky in the sky.
man standing on the beach near sea	man standing on the beach standing on the beach on the city beach.	man standing on the beach standing at the beach on the beach beach.	man standing on the beach standing on the beach standing at the city beach.	man standing on the beach standing on the beach on the beach beach.
	man standing on the beach standing on the beach near the beach.	man standing on the beach standing on the beach standing on the beach at the beach.	man standing on the beach standing on the beach on the beach.	man standing on the beach standing on the beach at the beach
grand nebula in the universe	star giant birds in the huge galaxy circle the planet in the cloud	giant star wars in the beautiful moon of the giant symphony in the earth.	huge star in the crystal world in the big stadium	giant star the great moon in the a huge galaxy in the giant field.
	giant star stars in the huge moon the a huge field in the beautiful city.	the giant bird in the giant the huge world in the moon shaped.	giant star the moon in the huge galaxy field in the giant field.	giant birds the world in the beautiful star planet in the stadium.
realistic scenery of Houston texas city view under a starry sky in hyperrealistic style and ultra HD, 8k	dramatic urban landscape of a city construction photo with city marquee and cloudy buildings in the foreground, a bright buildings movie studio imaging the scene in	beautiful skyscaw electronic photos star crystal-lit city skyline of a city scene in a scenic highway with a bright city skyline, home cabin on	a dark-park high-lit city skyline scene behind huge city laid buildings and landscapes, scoredReport artist dressed in the lovely backdrop,	downtown drone shot sharply with a very bright and colorful skyline city background, a backdrop with technical snowy skyline city skyline, the musical play a
	a competitive picture through a bright city cinema camera with a massive neon, big city mountains and city skyline buildings in the	a scenic city photo photographs show extreme bright sunshine, a hotel lights exciting city a skyline city at a peak, sky construction big city, with	city photography studio Kill a beautiful cityscape shot up close and with an urban theater star skyscrap backdrop with a city skyline square, the	scenic blue skysct star buildings with city and city skyline aerial view in the foreground in the background in a sunny bright urban setting overlooking
a pink car	a pink car a car	a pink car	a pink car a car	a pink car a car
	a pink car	a pink car	a pink car a pink car	a pink car a pink car
a handsome-looking horse rider	a well-dressed man a smiling horse riding a horse.	a well a handsome horse-m a horse smiling. a handsome horse man riding a horse.	a very a handsome a horse man dressed a his horse riding his horse	a very a tall a horse-ring horse rider a horse riding his short a horse.
	a well-eired a man a smiling horse riding a horse rider.	a well a- dressed horse wearing a happy horse riding a horse rider. a horse. a good horse. a horse	a well-looking a young horse-hired man riding a horse.	a handsome a horse- aired man riding a attractive horse.

Table 1: This table shows the performance of VD on text-variation.

5. Limitation

So far, we have shown that VD is a powerful model with outstanding capacities. However, VD still has noticeable limitations in some aspects, such as image-to-text, *etc*. In this session, we would like to discuss the limitation of our work as well as future research directions for improvements.

Limited Latent Space: During our experiment, we noticed that VD's major weakness is text generation (*i.e.* image-to-text, text-variation, I2T2I). We believe that such weakness can be largely improved if we expand the scope and capacity of the Optimus VAE [7]. Unlike the AutoKL [11] (*i.e.* the image VAE), which transforms images into latent features with dimension $4 \times 64 \times 64$, and thus keeps the necessary spatial information, Optimus's latent vectors are 768 single dimension vectors generated using Bert [2] which may be inadequate for long text sentences. We believe that a better solution should adapt word locations and orders, forming a latent space of sequences, which makes text generation and restoration in later stages easier. In our experiment, we also noticed that VD tended to generate sentences with repeated descriptions, which partially proves our guess that the corresponding VAE is weak in understanding word locations and orders.

Imperfect Data: Another issue that limited VD's performance was the imperfect text data we used in training. As mentioned in the main article Session 4.1, we formalized web-scraped prompts and captions with extensive engineering effort. We notice that with these cleaned captions, VD's training procedure on text-generation tasks had become more robust and easier to converge. Therefore, an immediate future target for VD research is to prepare a finetuned dataset that helps

VD improve its model accuracy. Meanwhile, the pretrained Optimus VAE also had certain data limits. In our experiment, we noticed that Optimus VAE had difficulty reconstructing Laion2B captions. An example is shown below: "*Assorted Cuff Colors Sandals Platform Leather Unique Ladies 1TO9 Cow Camel Ankle Brown EqBF6TT0*" (Laion2B), "*leatherback canvas females posses exotic fruits terme white grass patchions*" (Optimus Reconstruction). We think the issue comes from a domain shift from Optimus's training data (*i.e*. PTB [8], SNLI [1], Yahoo and Yelp corpora [14, 4]) to VD's training data (*i.e*. Laion2B [13]), in which the former contains normal sentences with good grammar, and the latter contains long and descriptive sentences with online freestyle. Therefore, preparing a finetuned text VAE could be the critical next step for future VD research.

6. Gallery

Please see Figure 12, 13, 14, 15

References

[1] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015. 13

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 12

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 5

[4] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*, 2019. 13

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 7

[6] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023. 7

[7] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. *arXiv preprint arXiv:2004.04092*, 2020. 12

[8] Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Using Large Corpora*, 273, 1994. 13

[9] 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. 7

[10] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 7

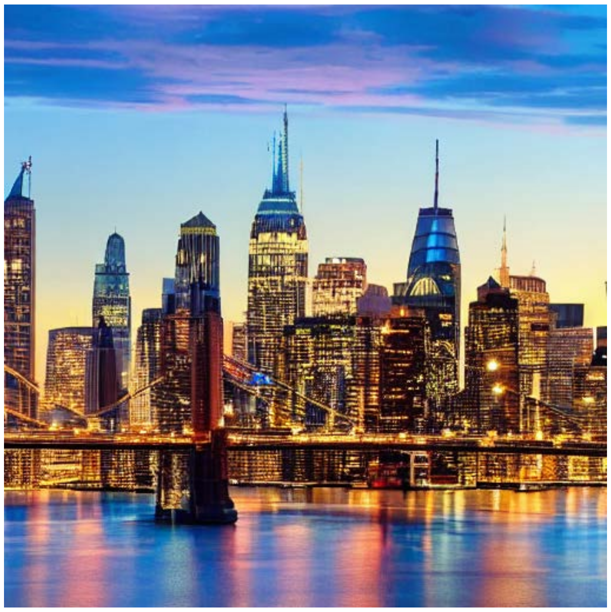
[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 7, 12

[12] 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, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 7

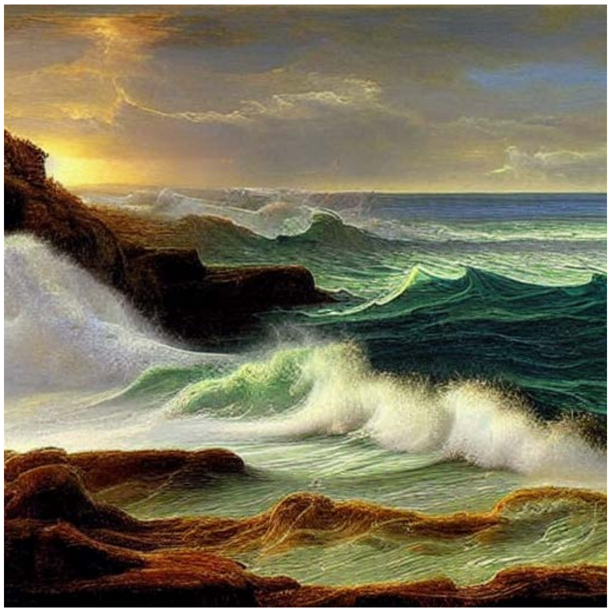
[13] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 13

[14] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *International conference on machine learning*, pages 3881–3890. PMLR, 2017. 13

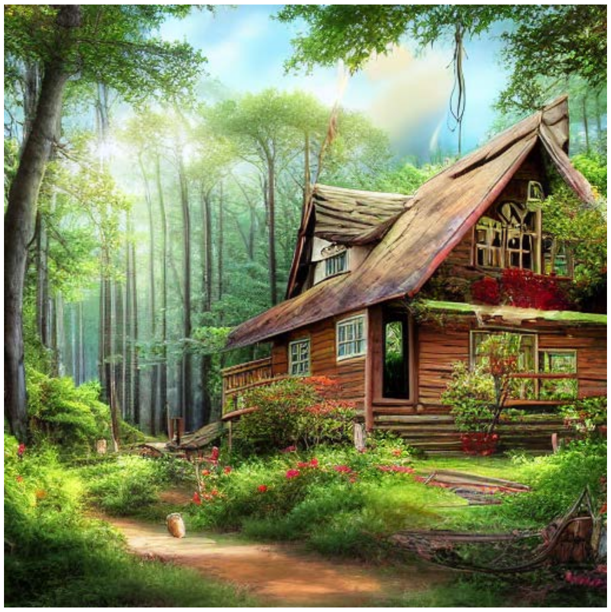
[15] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 7



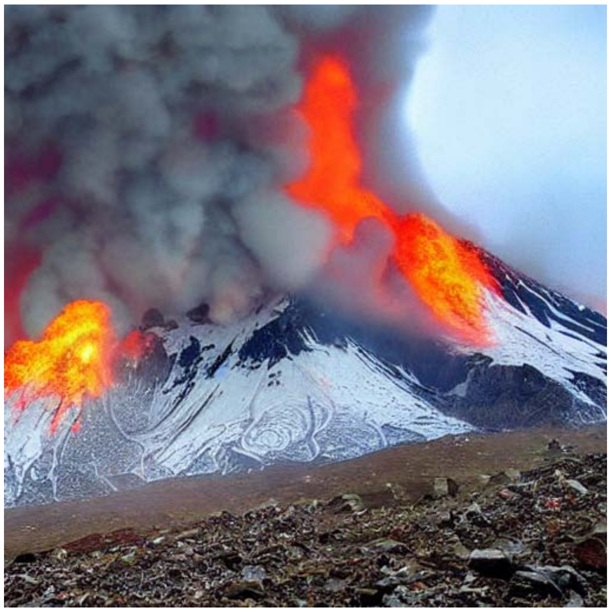
“A wonderful evening in New York City with a great view of Brooklyn Bridge and a magnificent city view of Manhattan, HD 8K”



“A beautiful painting of waves crashing on a cliff by Thomas Cole”

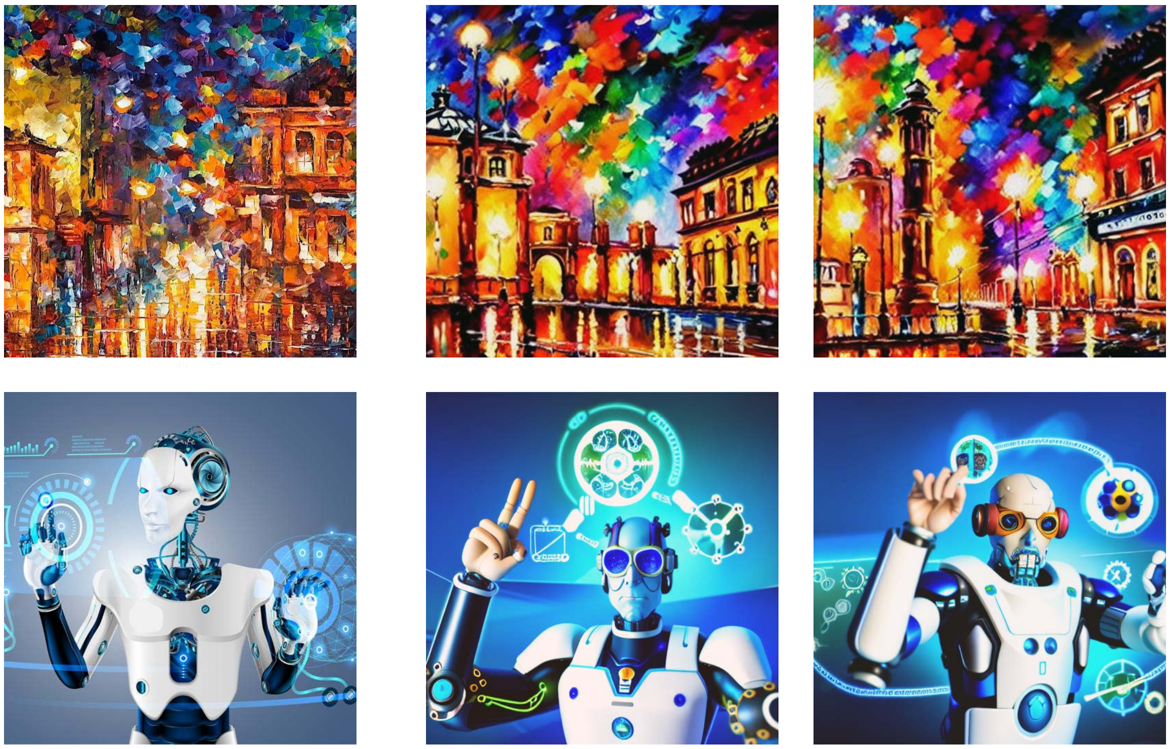


“Cluttered house in the woods anime oil painting high-resolution cottagecore Ghibli inspired 4k”



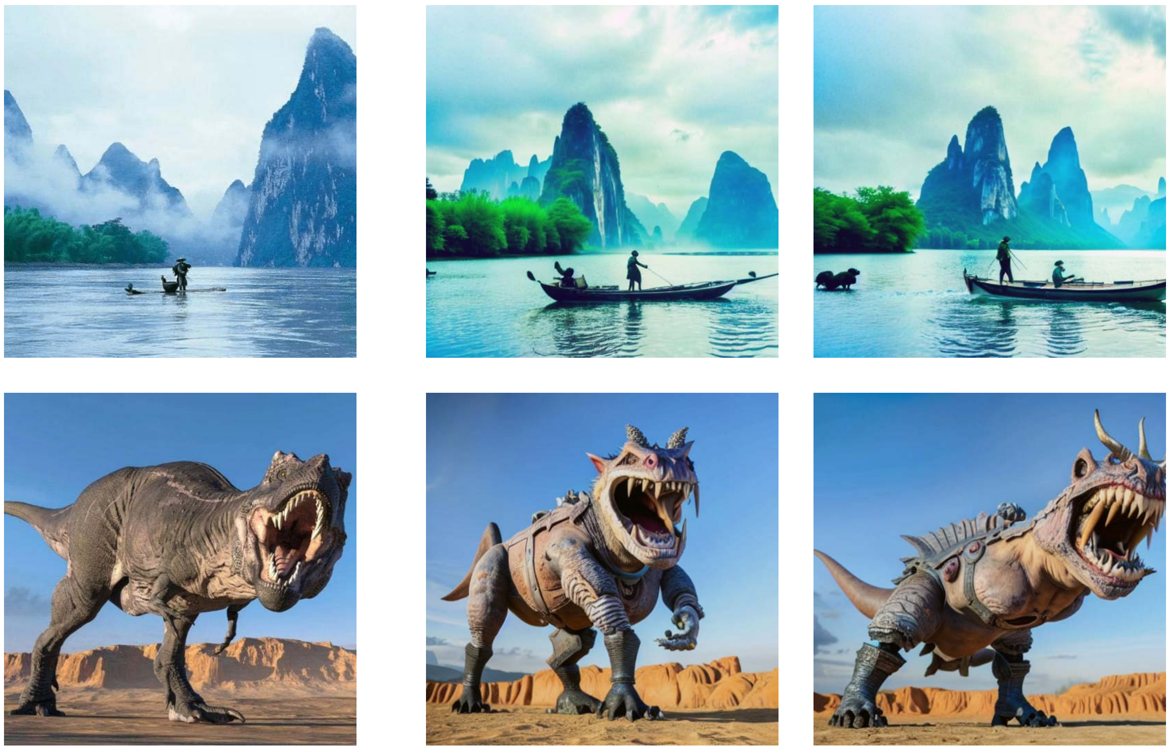
“Mountain Everest turns into an active volcano with hot lava coming out from the ground”

Figure 12: More text-to-image results.



Input Variation #1 Variation #2

Figure 13: More image-variation results.



Input Variation #1 Variation #2

Figure 14: More image-variation results with some mild semantic focus to achieve better photorealism.

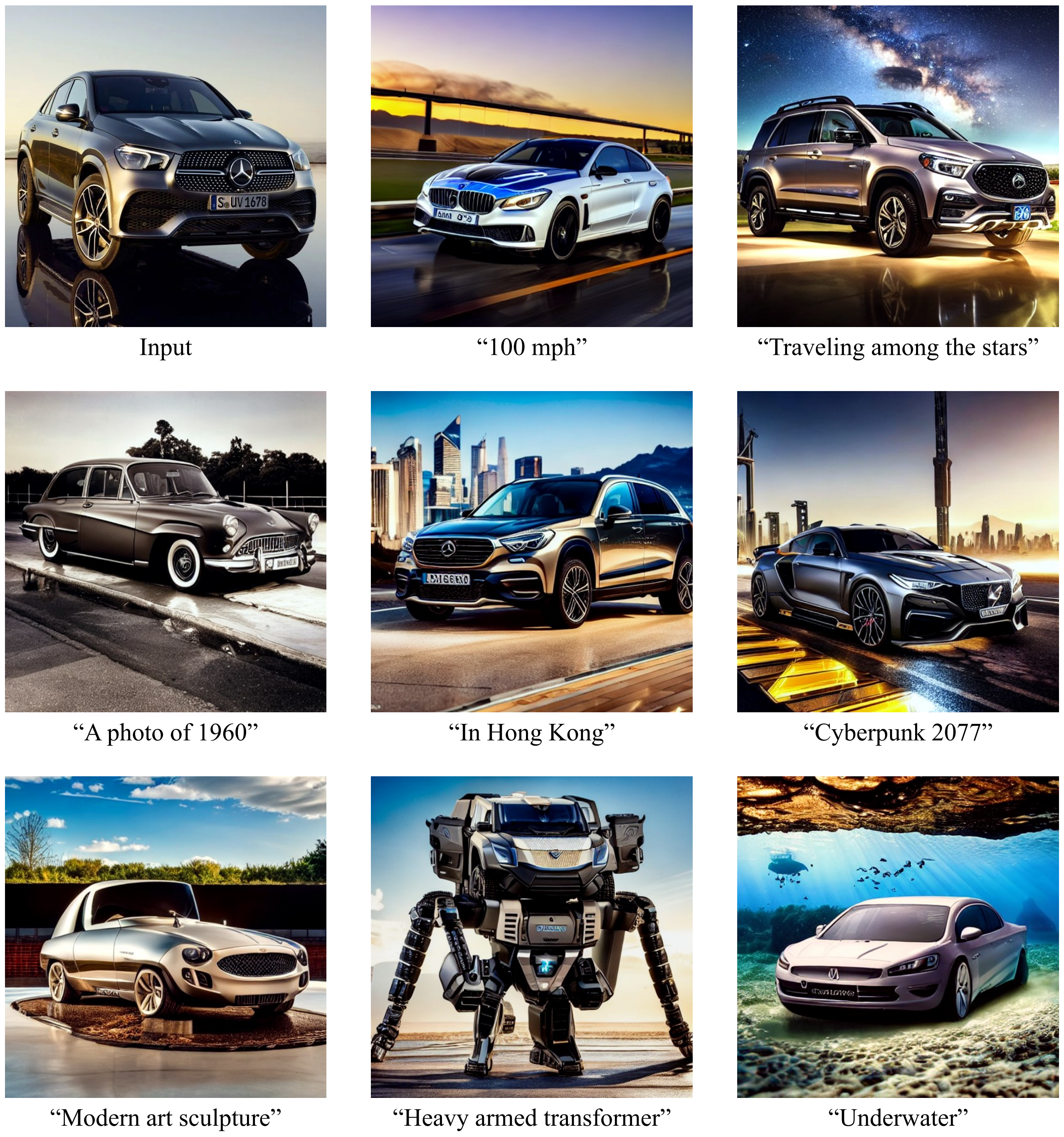


Figure 15: More dual-context blender results, in which the input image is shown at the top left corner, and input prompts are shown as sample labels.