Coupled Diffusion Sampling for Training-Free Multi-View Image Editing
– Supplementary Material –

A. Expanded theoretical analysis

A.1. Marginal distributions and coupling

Consider distributions A, B on the same sample space $\mathcal{X}$ with density functions p_A, p_B. We are interested in the effects of applying the coupling term $U(x^A, x^B)$ in coupled DDPM sampling. For clarity in this section, we will separate the coupling strength λ from U which we will use to denote the potential function to minimize. Note that when modeling a distribution with a diffusion model, we predict the score $s(x) = \nabla_x \log p(x)$, so when we add the coupling term, the log-density for the joint density becomes

$$\log p^{AB}(x^A, x^B) = \log p^A(x^A) + \log p^B(x^B) - \lambda U(x^A, x^B) + C, \tag{1}$$

for a constant C. By exponentiating, we find

$$p^{AB}(x^A, x^B) \propto \exp(\log p^A(x^A) + \log p^B(x^B) - \lambda U(x^A, x^B)) \tag{2}$$

$$= p^A(x^A)p^B(x^B)\exp(-\lambda U(x^A, x^B)). \tag{3}$$

To obtain the marginal p^A_λ for distribution A after coupling, we can integrate over x^B, and get

$$p^A_\lambda(x^A) = \frac{p^A(x^A)}{Z}\int p^B(x^B)\exp(-\lambda U(x^A, x^B))dx_B, \tag{4}$$

for a normalization constant Z. For our choice of U as the euclidean distance, we have $U = \frac{1}{2}\|x^A - x^B\|^2$. So we can define a Gaussian kernel $k_\lambda(x^B - x^A) \propto \exp(\frac{\lambda}{2}\|x^A - x^B\|^2)$, and then rewrite the density as

$$p^A_\lambda(x^A) = p^A(x^A)(p^B * k_\lambda)(x^A) \cdot \frac{1}{Z}. \tag{5}$$

Remark: The formula shows that the marginal for A is exactly the original p_A but reweighted pointwise with a kernel-smooth density estimate of p^B, and the smoothing scale is $\frac{1}{\lambda}$. Intuitively, as $\lambda \to 0$, we recover the original marginal since the kernel smoothing becomes very broad and effectively constant. On the other hand, as $\lambda \to \infty$, where we have a very strong coupling, the variance of the kernel approaches zero. In that case, we model the product distribution $p^{AB}_\lambda(x) \propto p^A(x)p^B(x)$. Note that this indicates that prior work that attempts to model the joint distribution [4, 12] reduces to a special case of our framework for infinity large λ.

A.2. KL divergence and training distribution

To ensure that the diffusion model produces valid samples within the distribution, we need to ensure that we minimize the deviation from the training distribution. While analyzing the generalization of diffusion models within and outside the training data remains an active area of research [17], we can set an upper bound for the KL divergence between the typical DDPM step and our coupled DDPM step. This is possible because the intermediate noisy latents can be modeled as a "clean sample" corrupted by noise with known variance. The distribution of an intermediate noisy latent x_t is

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t I)). \tag{6}$$

Note that during sampling, we do not have access to x_0 and only the estimated $\hat{x}_0$. Given the current prediction, we sample from a gaussian distribution computed based on the estimated mean. When the shift δ induces a change in the mean of the noisy-latent distribution, the KL divergence between the original and new distributions can be computed as

$$\mathrm{KL}(\mathcal{N}(\mu, \sigma^2 I) \| \mathcal{N}(\mu + \delta, \sigma^2 I)) = \frac{1}{2} \delta^T \frac{1}{1 - \bar{\alpha}_t} I \delta \tag{7}$$

$$= \frac{\|\delta\|^2}{2(1 - \bar{\alpha}_t)}. \tag{8}$$

In our setup, δ is the coupling term which consists of the difference between the estimated clean latents scaled by the noise variance $\lambda(\sqrt{1 - \bar{\alpha}_t}) \nabla_{x_t} U$. Note that the KL divergence is proportional to the inverse of the noise variance, and by scaling the coupling strength by the noise variance, we cancel out the dependence of the KL divergence on the denoising timestep. This also aligns with the established intuition that over time, the generated images become more defined and the uncertainty decreases, so the sample steering should decrease accordingly. Continuing the analysis, by setting an upper bound of 2 for each entry in δ, and setting M to be the number of elements, we find that the KL divergence is bounded as

$$\mathrm{KL}(x_t \| x_t + \delta) = \frac{\|\delta\|^2}{2(1 - \bar{\alpha}_t)} \tag{9}$$

$$= \frac{\|\lambda(\sqrt{1 - \bar{\alpha}_t}) \nabla_{x_t} U\|^2}{2(1 - \bar{\alpha}_t)} \tag{10}$$

$$= \frac{\|\lambda \nabla_{x_t} U\|^2}{2} \tag{11}$$

$$\leq \frac{\|2\lambda M\|^2}{2} \tag{12}$$

$$= 2\lambda^2 M^2, \tag{13}$$

so a possible interpretation of the coupling strength λ is adjusting the upper bound on the KL divergence between the inference distribution and sampling distribution. In Fig. 2 we show that decaying coupling guidance is superior to maintaining a constant guidance strength across all denoising steps, which causes a degradation in performance and blurry results.

B. Visualization of the 2D and MV Coupling

Our pipeline consists of two steps: First, we run the 2D model on a single image to produce an edited reference. Second, we use the edited reference to condition SVC to perform image-to-MV, and couple it with samples from the 2D model to ensure faithfulness to the identity of the input images. In Fig. 1 we visualize the process.

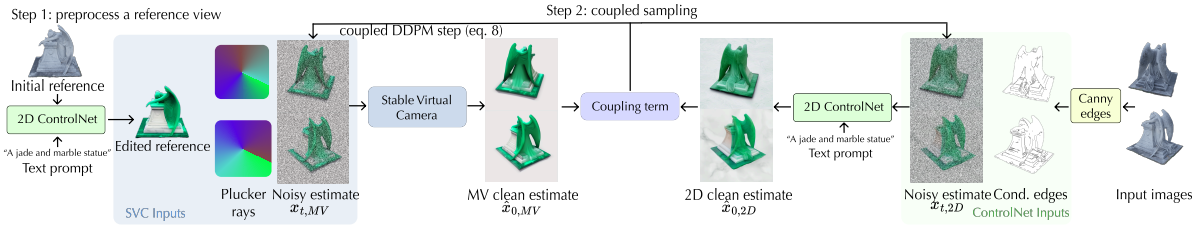


Figure 1. **Illustration of the coupling between 2D stylization and multi-view generation.** First, we start by editing a reference view to use as a condition for SVC. Second, we perform coupled DDPM sampling using the two models, and set $x_{0,MV}$ as our final output.

C. Additional discussion of limitations

While the proposed method offers a simple and efficient framework for multi-view consistent image editing, there are inherent limitations. Our method is a form of diffusion sampling guidance, so naturally the results quality and consistency depends on the base model used.. For example, when applying guidance to a multi-view model [8, 16, 24], if the model consistency is imperfect without guidance, it would remain imperfect with guidance. This can be observed with multi-view output as samples that show flicker in the background or object for example. Furthermore, the number of frames used depends on the

number of frames supported by the base model (e.g. 4 frames for MVDream [16], 6 frames for MV-Adapter [8], 21 frames for Stable Virtual Camera [24], and 81 frames for videos with Wan [19]).
Furthermore, similar to prior test-time guidance methods [2, 5, 7], the best results are produced using the highest possible guidance before the results collapse. The tolerated guidance values depend on both models used, so for each dataset, a different guidance values might be needed. However, since our method is training-free, cheap, and fast, it is easy to test varying numbers of hyperparameters to select from.
In section K, we show the effects of coupling in pixel-space of models with different latent spaces. However, in most of our experiments, we focus on settings where the 2D editing and multi-view models share the same latent space. The focus of our work is to show how we can enable Multi-view editing *without* the need for expensive and scarce paired 3D editing data. We show that simply a strong generative multi-view model, can be used in combination with a 2D model to perform the edit. It is always possible to train models with new latent spaces with the existing data. On the other hand, collecting new 3D paired data is a distinct significant challenge. That being said, we believe that our work paves the path to more general frameworks that operate in significantly distinct latent spaces, potentially by using lightweight latents alignment modules.

D. Implementation details

As our primary multi-view generation base model, we use Stable-Virtual-Camera (SVC) which is trained to process 21 frames at once, and use one or several consistent images for novel view synthesis. As we would like to edit a collection of views, we do not have access to more than one consistent *edited* photos, since we can only edit one image at a time with the 2D editing model. In our experiments, we edit a reference image, and then use it as the conditioning view. Note that this conditioning view has great influence on the outcome, as it dictates the distribution of acceptable 3D scenes that SVC would synthesize.
We transform stable-virtual-camera into DDPM by converting the EDM based sampler into a DDPM scheduled by computing converting the noise levels into the appropriate alphas. Afterwards, since SVC was trained with a shifted noise schedule compared to SD2.1 image models, we re-align SVC's schedule with the 2D model's schedule for the coupled sampling to be effective.
We conduct our experiments using NVIDIA A6000 GPUs. As our approach only requires a feed forward pass, the memory requirement is equivalent to the combined memory of the two models used. A better memory utilization can be further achieved by loading and off-loading the models from the GPU, as we can run them sequentially and then compute the coupling term. We use 50 denoising steps for spatial editing and stylization, and 100 denoising steps with Neural Gaffer relighting. The runtime of the sampling process is 130 seconds using our GPU resources for generating the full 21 frames sequence. Note that in contrast, prior test-time guidance methods [2, 5] require large number of re-noising steps to ensure quality, making the runtime as high as an hour for a single frame.
In the experiments with Neural-Gaffer, one challenge is that Neural-Gaffer is trained on 256x256 images. On the other hand, SVC was trained on 576x576 images. We found that SVC performs very poorly on images of that size, and neural-gaffer does not generalize to 512x512 images or larger. After experimenting with the models, we found that at resolution size of 384x384, both models perform reasonably well and adopt that for the neural-gaffer experiments.

E. Evaluation protocol

We evaluate our method across different tasks: spatial editing, relighting, and stylization. For spatial editing, we construct scenes using assets from both Sktechfab (with Creative Commons license) as well as the official Blender webpage. We construct and render the scenes using Blender. For each spatial edit, we render the source sequence, as well as ground truth sequence after applying the edit for quantitative evaluation. To construct the "coarse edit" that we feed to the 2D model [1], we use the depth maps associated with each image to unproject the image, apply the spatial transformation in 3D, and reproject the edit. Since we have multi-view inputs, the depth maps can be computed easily. To ensure accuracy for the pruproses of quantiative evaluations, we use GT depth maps rendered in Blender. In total, we create 9 spatial edits using 3 scenes. For stylization, we use 10 distinct objects sourced from Sketchfab, choosing objects with varying complexity. We use Blender to render a rotating trajectory around each object. Our trajectories demonstrate effectively demonstrate whether the coupled sampling can recover the identity of the input, as we ensure that there are unseen regions that are not observed in the conditioning input to the multi-view model. Since it is not possible to evaluate stylization around ground truth results, we utilize VBench [21] to evaluate the temporal and subject consistency, as well as the user study. For environment-map conditioned relighting, we use the objects sourced from Neural-Gaffer [9] for 3D relighting evaluation, and include two additional objects (an object with specular materials, and a diffuse object), and evaluate each object on 5 distinct target lightings. In total, we evaluate on 35 scenes.

Figure 2. **Comparing guidance schedule.** Here we show how decaying the guidance strength by the noise variance is effective, compared to using a constant coupling strength which causes blurry results. Here we use a text-to-image model, and perform coupling between two samples with distinct prompt. The result is also aligned with the intuition that over denoising time steps, the sample becomes more defined and perturbing the sample would cause a decay in performance.

We conduct the user study using the online platform *Prolific*. For each task, we include 9 questions in a *best-of-n* comparisons, where we show the user the input and all the methods, and ask the user to select the best output. We instruct the user to emphasize consistency, as well as faithfulness to the desired edit, depending on the task. In settings where we have Ground Truth sequence, we include the Ground Truth to the user as a reference. Each question is answered by 25 users.

F. Implementation details of baselines

Here we cover the details for implementing the baselines across our editing tasks. In our setup, the mutli-view diffusion model used [24] requires a conditioning image as an input, so we start by using the 2D editing model to sample an edited frame to use it as a condition. When using the baselines that rely on composing diffusion models [4, 12], we ensure that the multi-view model is conditioned on the exact same edited image for fairness and to clearly highlight the distinction between our method and the baselines. We implement Liu *et al*. [12] by composing the scores of the two diffusion models by linearly combining the scores for each step, and having the two models jointly denoise a shared sample. We implement Du *et al*. [4], by composing scores similarly to Liu *et al*. [12], with combination of 3 Langevin MCMC steps (which uses additional compute compared to our method). The purpose of the Langevin MCMC steps is to increase robustness for the samples to remain within distribution. In our case, we still show that the prior approaches suffer from inconsistencies compared to our coupled sampling approach. We also include the outputs of Stable Virtual Camera [24] on Image-to-MV generation when using the same conditioning edit used with the above baselines, and we include the per-image 2D editing (using the 2D model chosen for coupling for the specific task).

For stylization, we further use TEXTure [15] and Hunyuan3D [18]. For both of those baselines, we use the GT mesh associated with the object we are interested in stylizing. TEXTure relies on SDS for stylization, while Hunyuan3D uses a specialized multi-view generation model to synthesize consistent texture. For completion, in the supplementary webpage we also include additional image-to-3D baselines: Trellis [20] which generates a 3D asset conditioned on an image, and Tailor3D [14] which relies on editing the front and back of an object, then using the edited images to generate a 3D asset. Both of these approaches generate consistent 3D results (as they generate a 3D asset), but suffer in the faithfulness to the input image. Furthermore, since Tailor3D [14] edits two images independently, the conditioning edits used may not be consistent with each other.

For spatial editing, we rely on SDEdit [13] for using the coarse edit as a starting point as well as a conditioned edited image, and generate the multi-view sequence. To improve the baseline performance, we also insure that we condition on frames with

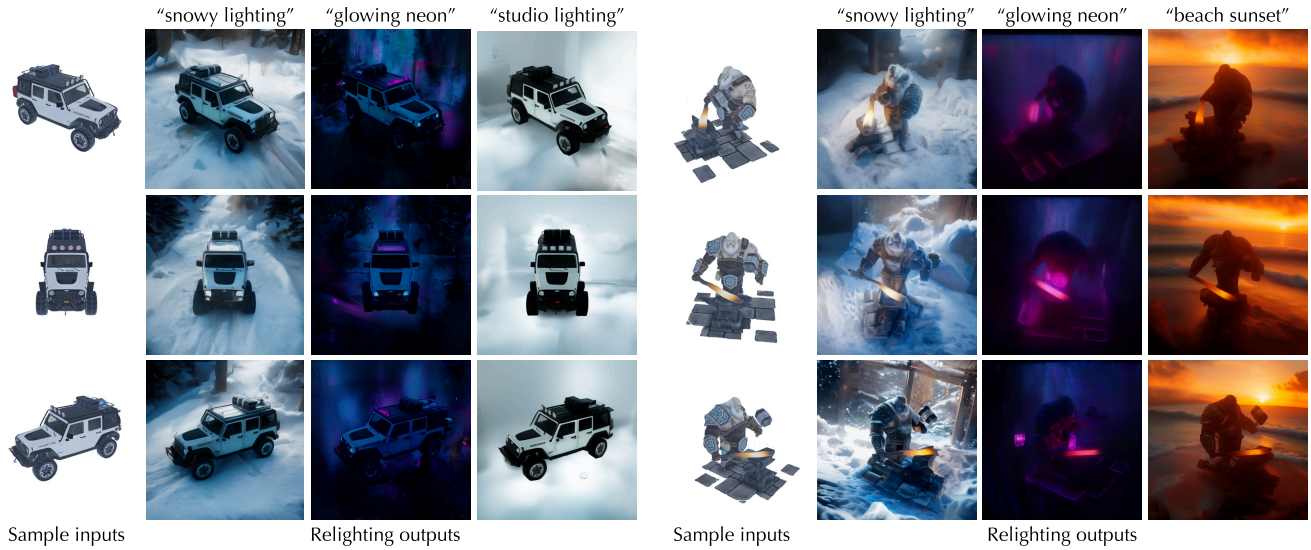


Figure 3. **Text based relighting.** We combine IC-Light [23], which enables text-based relighting with stable virtual camera to obtain multi-view results.

consistent inpainted regions (since the coarse edit may contain holes), to reduce the flickering when using SDEdit.
For environment map conditioned relighting, we include the NeRF based 3D relighting implemented in Neural-Gaffer [9]. We train a NeRF with densely rendered datasets of the object to obtain a high quality NeRF (note that our method does not access the densely rendered input), and then we incorporate Neural-Gaffer in the NeRF editing loop using their implementation. This is a strong baseline as it has access to ground truth radiance field. Note that a significant drawback is the need of NeRF optimization and editing, which can take between 30 minutes and an hour (in contrast, our method is feedforward, taking 130 seconds).
Below, we show the runtime of our method and baselines. Our performance is orders of magnitude faster than optimization-based methods.

Method	Ours	Liu *et al*. [30]	Du *et al*. [12]	TEXTure	Hunyuan3D	GT NeRF + NG	Instruct-N2N
Runtime (s)	38	38	84	107	29	1110	4500+

G. Text conditioned relighting

To show more drastic relighting outputs, we use IC-Light [23] for 2D relighting in combination with Stable Virtual Camera [24]. IC-Light operates by relighting the object and adding a suitable background conditioned on a text prompt. In Fig. 3 we show diverse multi-view relighting results using our method. Please refer to the supplementary webpage for additional video results.

H. Coupled Diffusion Sampling with Flow Models

In our experiments with coupling different prompts for text-to-image generation in Fig. 7 of the main paper is, we used Flux [10]. Note that Flux [10] is a flow model, so we need to adapt it to perform our coupled sampling. To sample from Flux using our proposed sampling method, first we transform the velocity $v_\theta(x_t)$ to the score function $s_\theta(x_t)$, as it can be linearly transformed into score functions via $s_\theta(x_t) = -\frac{-tv_\theta(x_t)+x_t}{1-t}$. Then transform the inference schedule to be DDPM via time reparameterization [11] by computing the appropriate alpha values that match the noise levels associated with each time step. We follow the same approach for coupling the video model Wan2.1 [19], since it is also a flow-based model.

H.1. Effects of Guidance Strength

As an additional illustration, in Fig. 4 we show how the samples change as we increase the coupling strength, while using the same initial random noise and randomness seed.

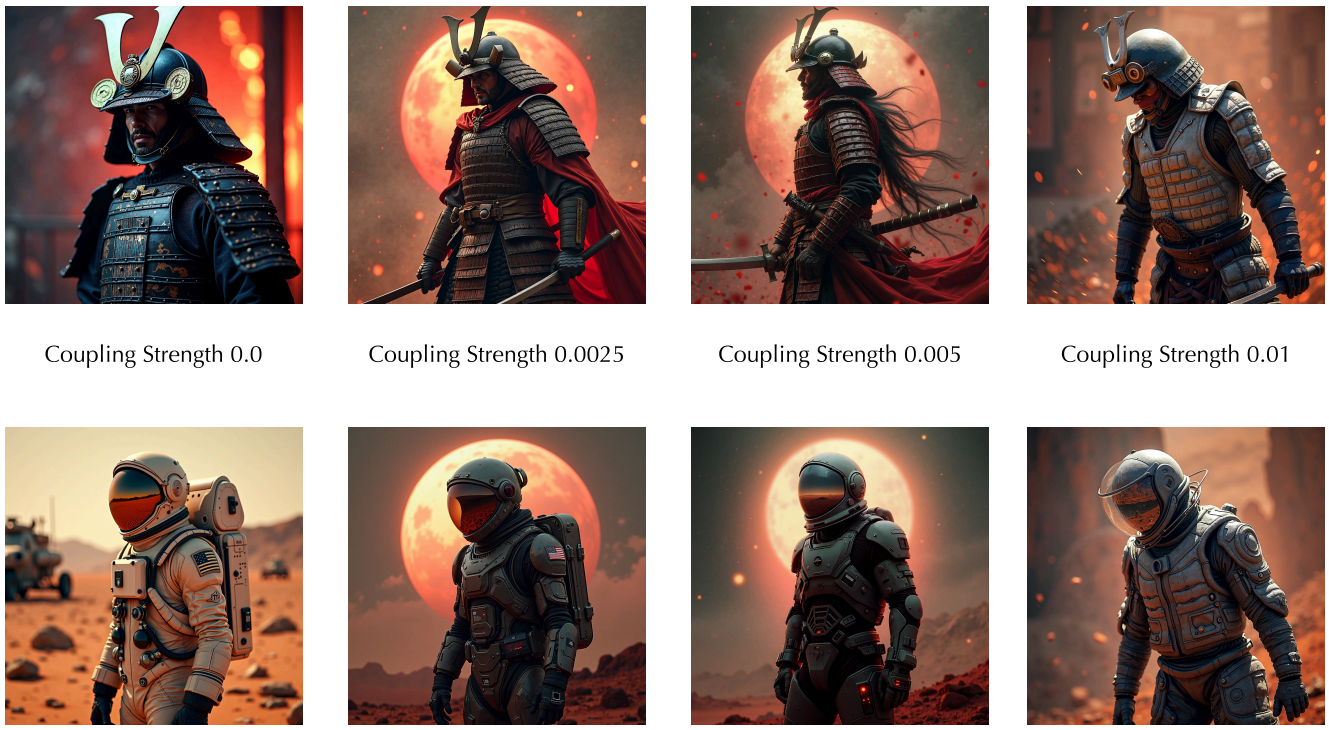


Figure 4. **Effects of coupling strength.** We illustrate the effects of coupling strength on the spatial alignment between samples as we vary the coupling strength while keeping the initial noise and random seed.

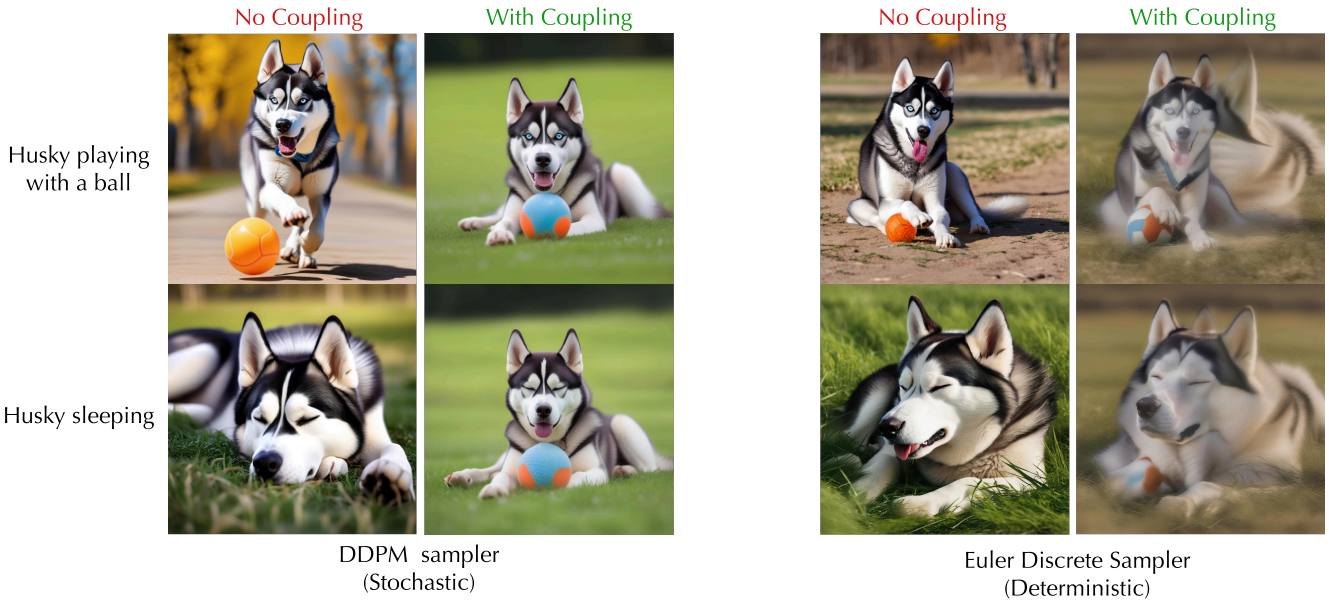


Figure 5. **Sampler comparison.** When using a stochastic sampler, the coupling can lead to natural guidance pulling the outputs towards each other. On the other hand, a deterministic sampler would simply output the average of both samples, as ODE based sampling does cannot recover from noisy guidance.

Figure 6. **Coupling in different multi-view models.** We implement coupling on T2I and T2MV models with two different backbones. We couple SD2.1 with MVDream [16], and SDXL with MVAdapter [8], which operates in SDXL latent space. In both cases, the coupled multiview samples show an increase in realism and a decrease in "objaverse" appearance.

I. Effects of stochasticity

One observation one would make is that our coupling term resembles linearly combining the intermediate samples of the two models, so one may wonder why we do not simply get images that are a linear average of the two outputs. Indeed, when we use a deterministic sampler, like Euler Discrete Sampler that's commonly used, this is the outcome that we encounter as we show in Fig. 5. However, when using a stochastic sampler like DDPM where noise is injected at every timestep, the model needs to correct for the added noise. When we include our coupling term in the stochastic step, the model can naturally correct or reject parts of the guidance that steers it away from its training distribution. This is also the reason we make our coupling term to be correlated with the noise level, by scaling it with $\sqrt{1-\alpha_t}$, since at step t, the model has the ability to correct for noise at that level, but steering the sample by a larger magnitude risks pulling the intermediate latents outside of the training distribution. Additionally, as intuitively understood about diffusion sampling, at the later time steps the structure of the outputs is already determined, so shifting the intermediate latents in a large direction can disrupt the sampling process.

J. Coupling Text-to-Image and Text-to-MV models

In the main paper, we presented multi-view editing results using Stable Virtual Camera [24]. Here, we further examine the impact of coupling on text-to-multi-view models, specifically MVDream [16], which extends Stable Diffusion 1.5 to produce four consistent views, and MV-Adapter [8], which leverages the more advanced SDXL backbone and operates in the SDXL latent space. For coupling, we use SD1.5 and SDXL as the respective text-to-image models. As shown in Figure 6, text-to-multi-view models often generate objects with a CGI-like appearance, likely due to their training on datasets such as Objaverse [3]. Introducing our coupling approach encourages the multi-view samples to better resemble real images, as modeled by the 2D diffusion models.
In Fig. 7 and Fig. 8, we highlight additional results from coupling text-to-multi-view models along with text-to-image models.

K. Pixel space coupling

In all previous experiments, we perform coupling in the latent space, as both models operated in the same latent space. When models operate in distinct latent spaces, the alternative is to decode the intermediate latent of each model to pixel space. Once we are in the shared pixel space, we can compute the coupling function, and use autograd to backpropagate the gradients

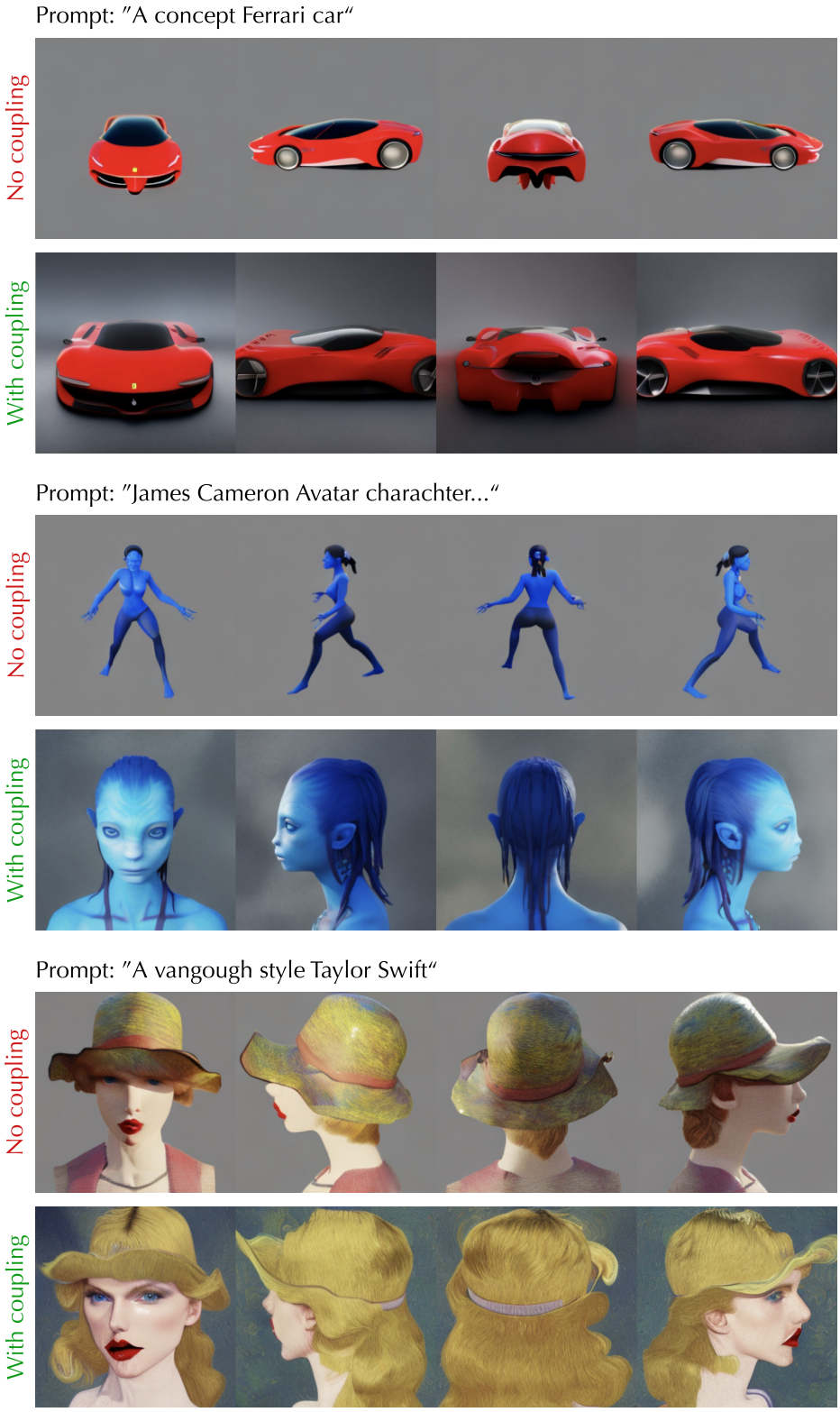


Figure 7. **Additional MVDream T2MV coupling results.** Here we show additional results on the output of Text-to-Multiview MVDream when coupled with Text-to-Image SD2.1.

Figure 8. **Additional MV-Adapter T2MV coupling results.** Here we show additional results on the output of Text-to-Multiview MV-Adapter when coupled with Text-to-Image SDXL.

through each model's decoder. In particular,

$$\nabla_{x^A} U(\text{Dec}_A(\hat{x}_0^A), \text{Dec}_B(\hat{x}_0^B)) \tag{14}$$

$$\nabla_{x^B} U(\text{Dec}_B(\hat{x}_0^B), \text{Dec}_A(\hat{x}_0^A)), \tag{15}$$

and then we can apply the coupling term to each latent.
In this experiment, we use Text-to-Image SD2.1, and use MVAdapter that is based on SDXL, so that the two models operate in distinct latent spaces. In Fig. 9 we show the results of coupling in pixel space.

L. Applications with MV-Adapter

One of the limitations of MV-Adapter is that it can only generate fixed set of camera views, making its utility for editing limited. Nonetheless, we show that we can still use it by editing the outputs it produces by performing coupling with single-image editing models. In Fig. 10, we show an example of using MV-Adapter for stylization, and relighting.

M. Outputs of InstructNeRF2NeRF

When running InstructNeRF2NeRF on our input sequences used for stylization with the same number of frames as our method and other baselines (21 frames), we find that the radiance field completely collapses. This is likely due to NeRF's inability to gradually handle inconsistency with less dense camera coverage.

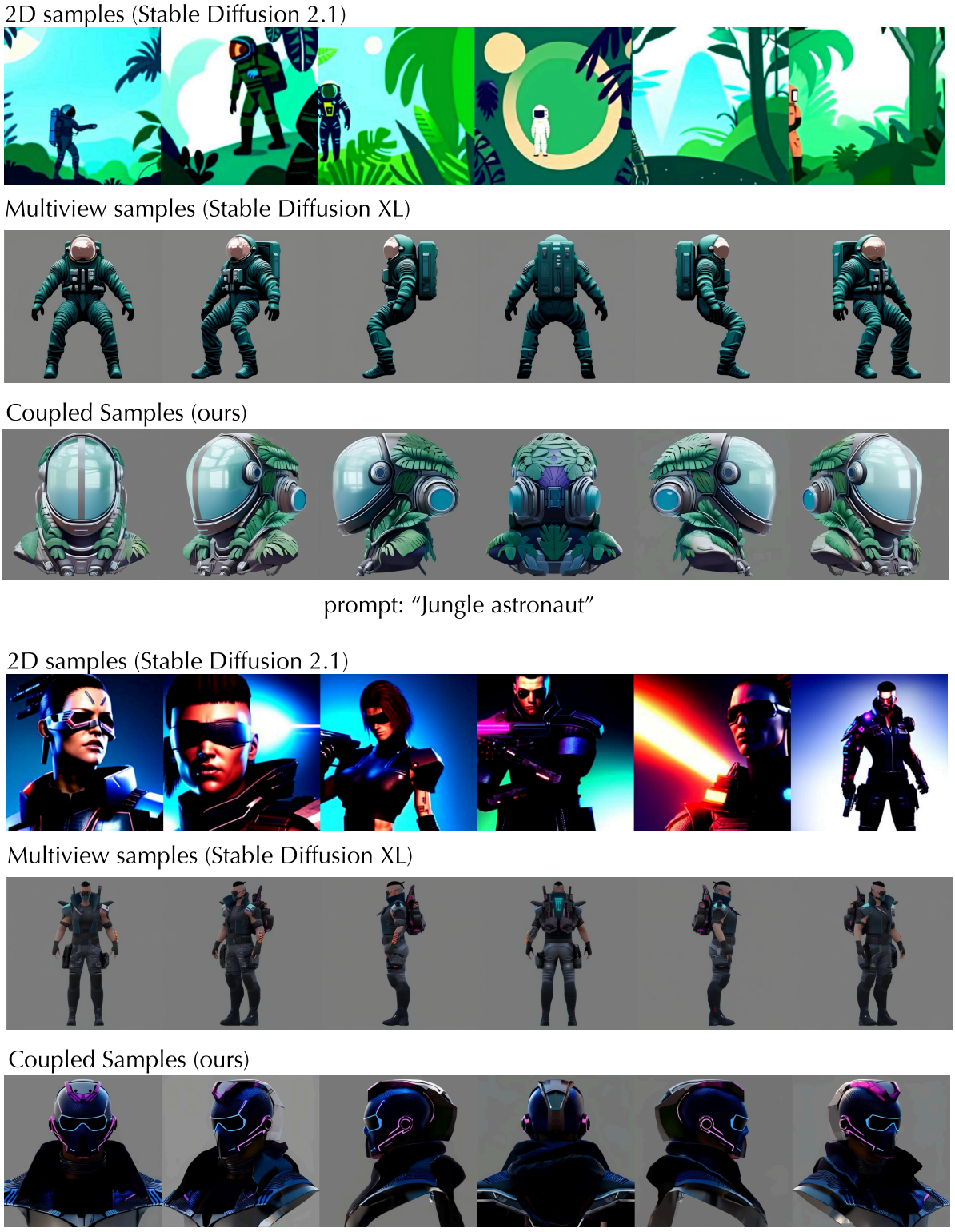
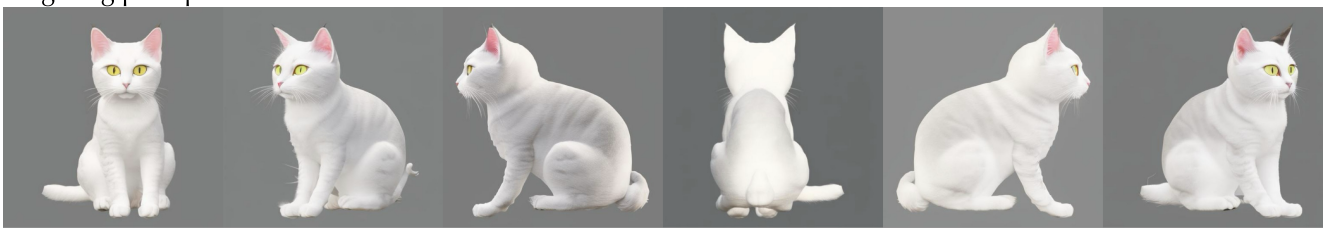


Figure 9. **Coupling in pixel space.** To show the effects of coupling when the two models operate in different latent spaces, we combine a Text-to-Image model (SD2.1 latent space), and Text-to-Multiview (SDXL latent space), and perform coupling in pixel space and backpropagating the gradients to the latents. Here we show the coupled multi-view samples that we get with our method, showing that our method can generalize to coupling models with different latent spaces.

Input views

Stylization (MV-Adapter + Control-Net)

relighting prompt: a snow white cat



Relighting (MV-Adapter + IC-Light)

relighting prompt: a white and orange tabby cat in studio lighting

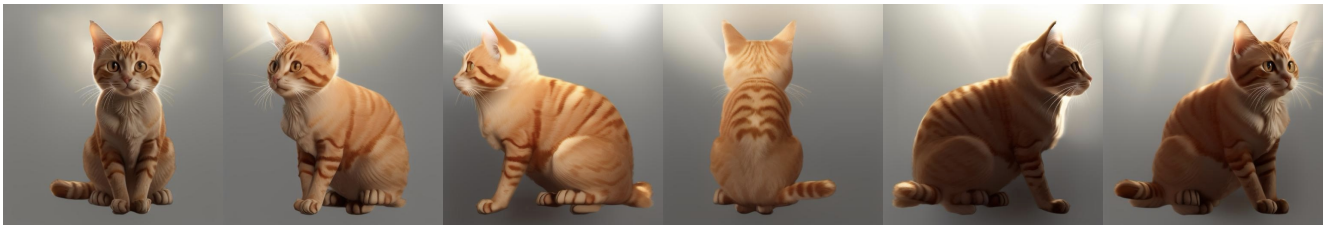
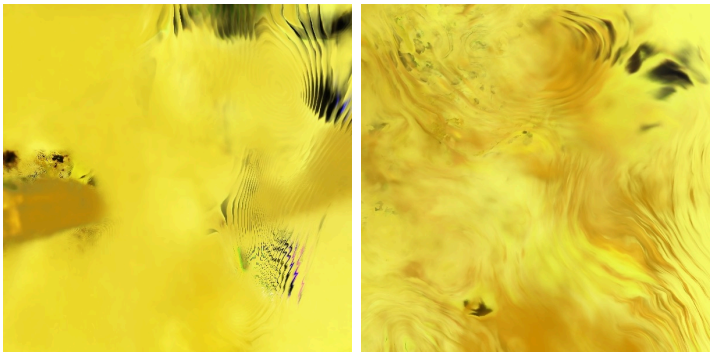


Figure 10. **Multiview editing with MV-Adapter.** Here we show editing results with MV-Adapter to achieve stylization by combining it with Control-Net [22] and relighting using IC-Light [23].



Sample input frame

+

Prompt: “make it a golden lamborghini”

InstructNeRF2NeRF renders

Figure 11. **InstructNeRF2NeRF outputs.** When running InstructNeRF2NeRF [6] on our input views, we find that the editing training loop with InstructPix2Pix completely collapses.

References

[1] Hadi Alzayer, Zhihao Xia, Xuaner (Cecilia) Zhang, Eli Shechtman, Jia-Bin Huang, and Michael Gharbi. Magic fixup: Streamlining photo editing by watching dynamic videos. *ACM Trans. Graph.*, 2025. 3

[2] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *ICLR*, 2024. 3

[3] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. 7

[4] Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. Reduce, reuse, recycle: compositional generation with energy-based diffusion models and mcmc. In *Int. Conf. Mach. Learn.*, 2023. 1, 4

[5] Daniel Geng and Andrew Owens. Motion guidance: Diffusion-based image editing with differentiable motion estimators. In *ICLR*, 2024. 3

[6] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *ICCV*, 2023. 12

[7] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 3

[8] Zehuan Huang, Yuanchen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. In *ICCV*, 2024. 2, 3, 7

[9] Haian Jin, Yuan Li, Fujun Luan, Yuanbo Xiangli, Sai Bi, Kai Zhang, Zexiang Xu, Jin Sun, and Noah Snavely. Neural gaffer: Relighting any object via diffusion. In *NeurIPS*, 2024. 3, 5

[10] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 5

[11] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024. 5

[12] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *ECCV*, 2022. 1, 4

[13] 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. In *ICLR*, 2022. 4

[14] Zhangyang Qi, Yunhan Yang, Mengchen Zhang, Long Xing, Xiaoyang Wu, Tong Wu, Dahua Lin, Xihui Liu, Jiaqi Wang, and Hengshuang Zhao. Tailor3d: Customized 3d assets editing and generation with dual-side images, 2024. 4

[15] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH Conference Proceedings*, 2023. 4

[16] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. In *ICLR*, 2024. 2, 3, 7

[17] Kiwhan Song, Jaeyeon Kim, Sitan Chen, Yilun Du, Sham Kakade, and Vincent Sitzmann. Selective underfitting in diffusion models. *arXiv preprint arXiv:2510.01378*, 2025. 1

[18] Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025. 4

[19] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 3, 5

[20] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *CVPR*, 2025. 4

[21] Fan Zhang, Shulin Tian, Ziqi Huang, Yu Qiao, and Ziwei Liu. Evaluation agent: Efficient and promptable evaluation framework for visual generative models. *arXiv preprint arXiv:2412.09645*, 2024. 3

[22] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 12

[23] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In *ICLR*, 2025. 5, 12

[24] Jensen (Jinghao) Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. In *ICCV*, 2025. 2, 3, 4, 5, 7