One-2-3-45++: Fast Single Image to 3D Objects with Consistent Multi-View Generation and 3D Diffusion —- Supplementary Materials

0. Table of Contents

This supplementary material includes:

1. Details of Evaluation Metrics

User Study: Our user study involved 53 participants, each of whom was tasked with comparing 75 pairs of results: 30 text-to-3D and 45 image-to-3D conversions. As illustrated in Figure 1, we presented each participant with pairs of results generated by two undisclosed distinct methods, alongside their corresponding input text or image. Participants were able to rotate and zoom in/out the 3D shapes to assess them from various viewpoints, subsequently choosing the one they deemed superior in quality and alignment with the input image or text. For each text-to-3D pair, an input text prompt was randomly selected from a set of 50 DreamFusion [15] prompts, and two different methods were applied randomly. Similarly, for each image-to-3D pair, an input image depicting a 3D shape was randomly selected from a pool of 1,030 GSO [5] shapes, with two different methods applied randomly. In total, 3,975 evaluated pairs were collected. We then calculated the preference rate for each method, both individually and in combination.

Other Metrics: For F-Score, we utilize a threshold of 0.05, with shapes normalized within a 0-1 range. For CLIP-Simlarity, we utilize CLIP ViT-L/14@336px. Since the predicted mesh may not have the same scale and pose as the ground-truth mesh, to ensure a fair comparison, we employ the following process to align the predicted mesh with the ground-truth mesh. First, we align the up direction for the results generated by each approach. Next, for each generated mesh, we perform a linear search over scales and rotation angles along the up direction. After applying each pair of scale and z-rotation, we utilize the Iterative Closest Point (ICP) algorithm to align the transformed mesh to the ground-truth mesh. Finally, we select the mesh with the largest number of inliers as the final alignment. This alignment process helps us establish a consistent reference frame for evaluating the predicted meshes across different approaches.

2. Details of Consistent Multi-View Generation

2.1. Consistency and Stability: Noise Schedule

The original noise schedule for Stable Diffusion, *i.e.*, the scaled-linear schedule, places emphasis on local details but has very few steps with a lower Signal-to-Noise Ratio (SNR), as shown in Fig. 2. These low SNR steps occur in the early denoising stage, which is crucial for determining the global low-frequency structure of the content. A reduced number of steps in this stage, either during training or inference, can lead to greater structural variation. While this setup is suitable for single-image generation, we have observed that it limits the model's ability to ensure global consistency between multiple views.

To empirically verify this, we perform a toy task by finetuning a LoRA [7] model on the Stable Diffusion 2 v-prediction model to overfit a blank white image given the prompt *a police car*. The results are presented in Fig. 3. Surprisingly, with the scaled-linear noise schedule, the LoRA model cannot overfit on this simple task; it only slightly whitened the image. In contrast, with the linear noise schedule, the LoRA model successfully generates a blank white image regardless of the prompt. While finetuning the full model may still be viable for the scaled-linear schedule, this example highlights the significant impact of the noise schedule on the model's ability to adapt to new global requirements.

As pointed out by Chen [2], high-resolution images appear less noisy compared to low-resolution images when subjected to the same absolute level of independent noise (see Fig. 2 in [2]). This phenomenon occurs because "higher resolution natural images tend to exhibit a higher degree of redundancy in (nearby) pixels, therefore less information is destroyed with the same level of independent noise." Consequently, we can interpret the use of lower resolution in Zero-1-to-3 training as a modification of the noise schedule, placing greater emphasis on the global require-

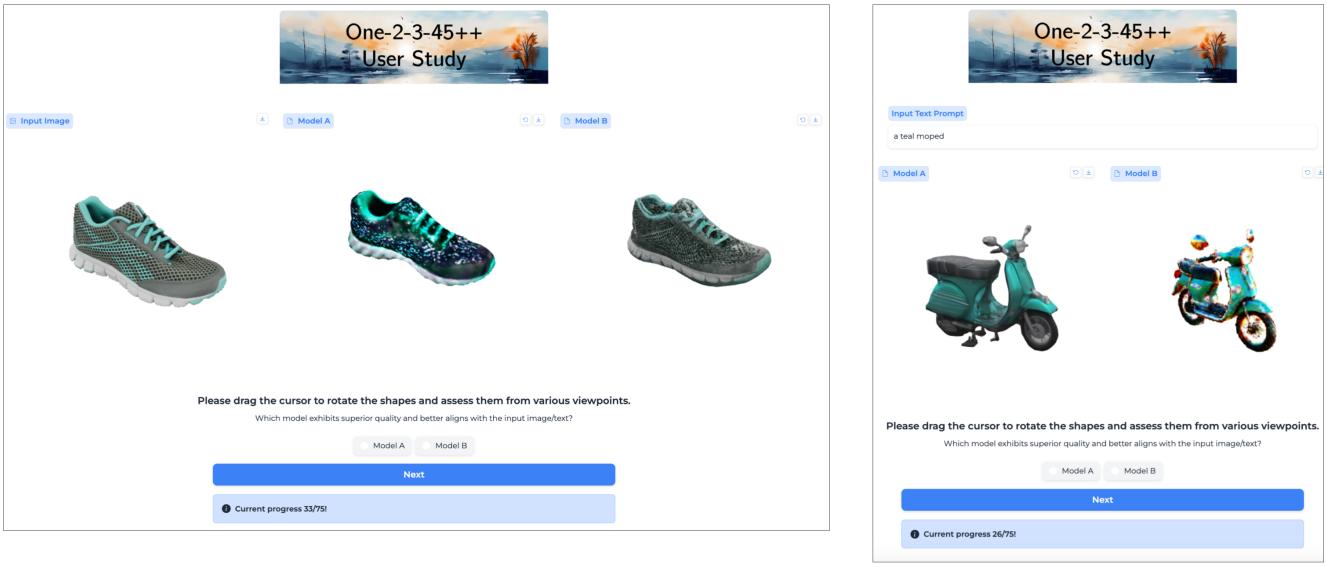
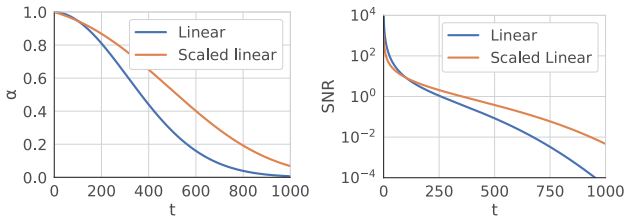


Image to 3D Text to 3D

Figure 1. A screenshot of our user study website is shown, where users can rotate and zoom in on the 3D models, then select the model that exhibits superior quality and better aligns with the input image or text.



(a) (b)

Figure 2. Comparison between the linear schedule and Stable Diffusion's scaled linear schedule.

Figure 3. **Importance of the noise schedule.** Noise schedule strongly affects the model's capability to adapt to new global requirements (generating a pure white image from the prompt *a police car* in this case). Notably, both schedules produce highly similar images before fine-tuning; therefore, we present only the result of the v model before fine-tuning.

ments of 3D-consistent multi-view generation. This also explains the instability issue of training Zero-1-to-3 with higher resolution [10].

In summary, we find it necessary to switch from the scaled-linear schedule to the linear schedule for noise in our model. However, this shift introduces another potential challenge: adapting the pretrained model to the new schedule. Fortunately, we have observed that the v-prediction model is quite robust when it comes to swapping the schedule, in contrast to the x_0- and ϵ-parameterizations, as illustrated in Figure 4. It is also theoretically supported that the v-prediction is inherently more stable [17]. Therefore, we have opted to utilize the Stable Diffusion 2 v-prediction model as our base model for fine-tuning.

Figure 4. **Swapping the noise schedule.** We swap the schedule of Stable Diffusion 2 v (Left) and ϵ-parameterized (Right) models from scaled-linear to linear at inference time without any finetuning. Prompt: *a blue clock with black numbers*. The ϵ-parameterized model exhibits a significant decrease in quality, while the v model produces a high-quality image with the linear noise schedule.

2.2. Local Condition: Scaled Reference Attention

In Zero-1-to-3, the conditioning image (single view input) is concatenated in the feature dimension with the noisy inputs to be denoised for local image conditioning. This imposes an incorrect pixel-wise spatial correspondence between the input and the target image.

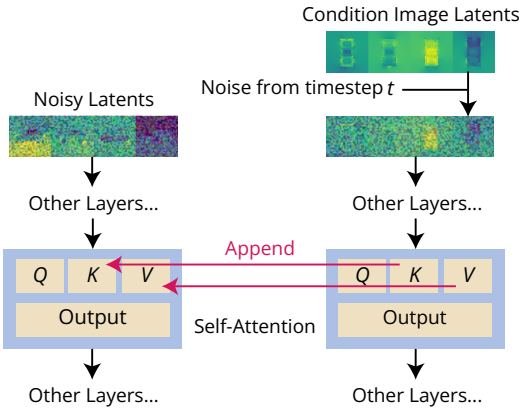


Figure 5. **Reference Attention.** It adds an additional conditioning branch and modifies key (K) and value (V) matrices of the self-attention layers to accept the extra condition image, which can fully reuse Stable Diffusion priors.

Figure 6. **Comparison on local conditioning.** We train the model with different levels of scaled reference attention on the ShapeNet Cars dataset. Output coherence with the input image is best on 5x scaled reference attention.

We propose to use a scaled version of Reference Attention to provide proper local conditioning input.

As shown in Fig. 5, Reference Attention [20] refers to the operation of running the denoising UNet model on an extra reference image and appending the self-attention key and value matrices from the reference image to the corresponding attention layers when denoising the model input. The same level of Gaussian noise as the denoising input is added to the reference image to allow the UNet to attend to relevant features for denoising at the current noise level.

Without any finetuning, Reference Attention is already capable of guiding the diffusion model to generate images that share similar semantic content and texture with the reference image. When finetuned, we observed that the Reference Attention works better when we scale the latent (before adding noise). In Figure 6, we provide a comparison from experiments conducted on ShapeNet Cars [1] to demonstrate that the model achieves the highest consistency with the conditioning image when the reference latent is scaled by a factor of 5.

2.3. Global Condition: FlexDiffuse

In the original Stable Diffusion, global conditioning comes solely from text embeddings. Stable Diffusion employs CLIP [16] as the text encoder and performs cross-attention between model latents and per-token CLIP text embeddings. As a result, we can make use of the alignment between CLIP image and text spaces to reuse the prior for global image conditioning.

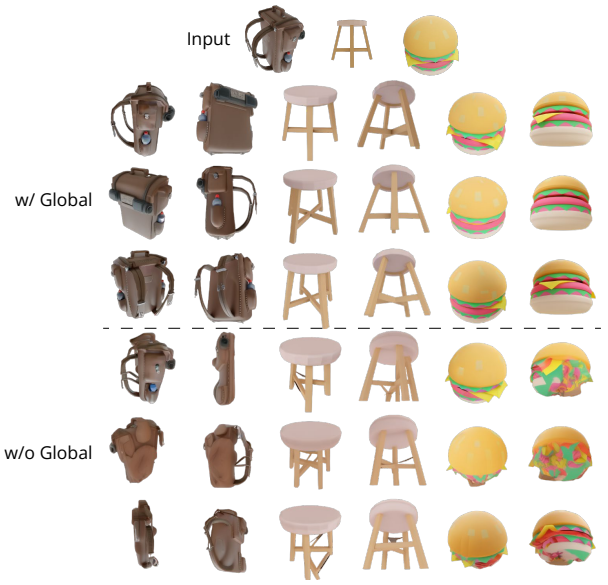


Figure 7. **Ablation on global conditioning.** In regions of the image that are not visible in the input, the results are significantly worse without global conditioning.

We propose a trainable variant of the linear guidance mechanism introduced in FlexDiffuse [18] to incorporate global image conditioning into the model while minimizing the extent of fine-tuning. We start from the original prompt embeddings T of shape $L \times D$ where L is the length of tokens and D is the dimension of token embeddings, and add the CLIP global image embedding I of shape D multiplied by a trainable set of global weights $\{w_i\}_{i=1,\dots,L}$ (a shared set of weights for all tokens) to the original prompt embeddings, or formally,

$$T'_i = T_i + w_i \cdot I, i = 1, 2, \dots, L. \quad (1)$$

We initialize the weights with FlexDiffuse's linear guidance:

$$w_i = \frac{i}{L}. \quad (2)$$

Since we do not impose any text conditions, so T is obtained by encoding an empty prompt. We present the results of trained models with and without global conditioning in Figure 7. In the absence of the proposed global conditioning, the quality of generated content remains satisfactory for visible regions corresponding to the input image. However, the generation quality significantly deteriorates for unseen

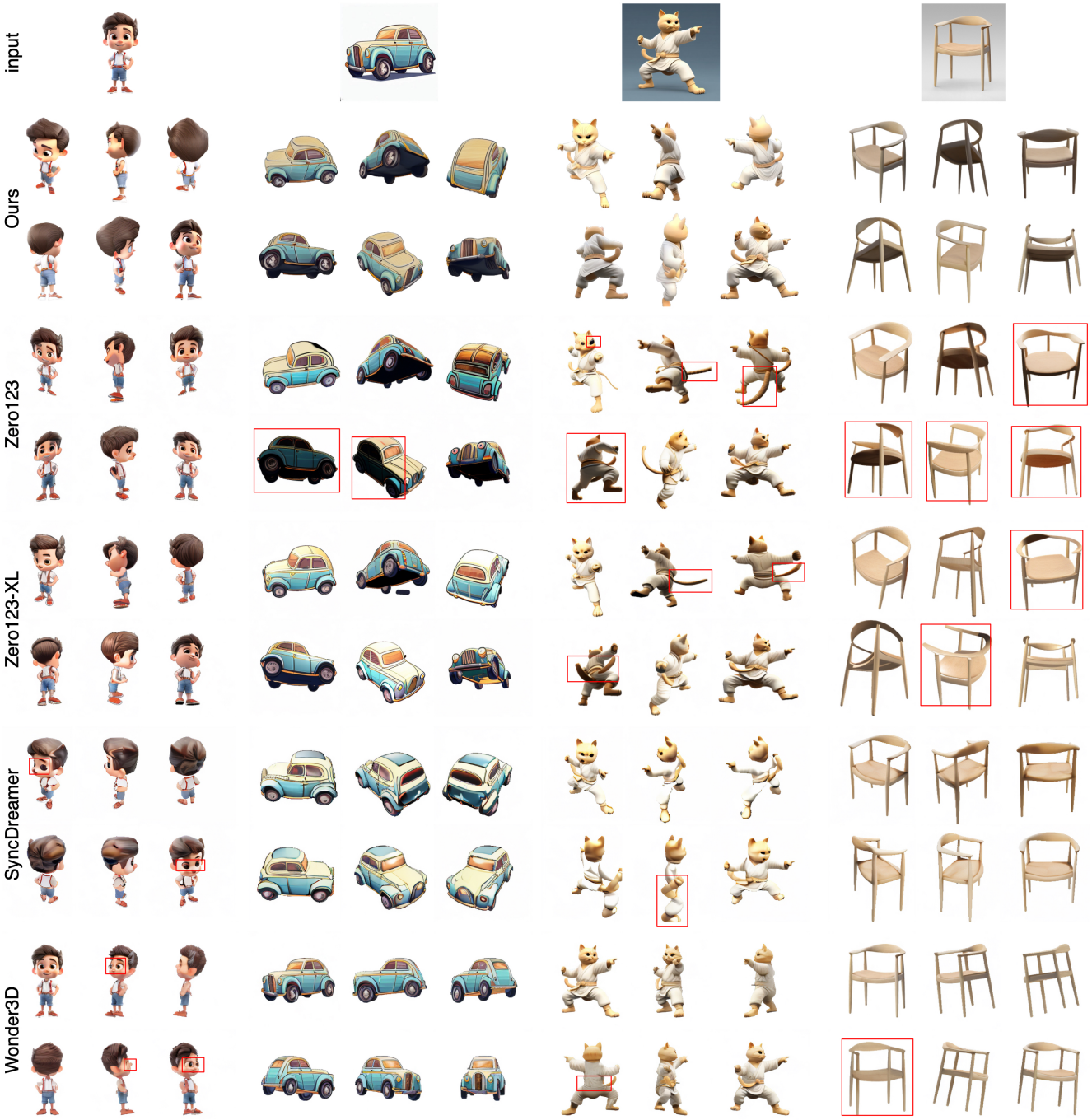


Figure 8. Multi-view prediction by different approaches.

Figure 9. Ablation study of texture refinement: In each pair, the left image displays the result before texture refinement, while the right image shows the result after refinement.

regions, as the model lacks the ability to infer the global semantics of the object.

2.4. Putting Everything Together

Starting from the Stable Diffusion 2 v-model, we train our model using all the techniques mentioned above. We train the model on Objaverse [4] data rendered with random HDRI environment lighting.

We adopt the phased training schedule from the Stable Diffusion Image Variations model [9] to further reduce the extent of finetuning and preserve as much prior in Stable Diffusion as possible. In the first phase, we only tune the self-attention layers and the KV matrices of cross-attention layers of Stable Diffusion. We use the AdamW[8, 14] optimizer with cosine annealing learning rate schedule peaking at 7×10^{-5} and 1000 warm-up steps. In the second phase, we employ a very conservative constant learning rate of 5×10^{-6} and 2000 warm-up steps to tune the full UNet. We employ the Min-SNR weighting strategy [6] to make the training process more efficient.

3. Details of 3D Diffusion

In the initial stage, the occupancy UNet is structured with five levels: 64^3, 32^3, 16^3, 8^3, and 4^3. These correspond to 32, 64, 128, 512, and 1024 channels, respectively. The global CLIP feature of the reference image is compressed using an MLP. These compressed features are then concatenated with the multi-view features and the original UNet features. For training, we utilize a batch size of 256 and a learning rate of 5e-4.

In the second stage, the UNet also has five levels, but with different resolutions: 128^3, 64^3, 32^3, 16^3, and 8^3, maintaining the same channel configuration as in the first stage. We utilize TorchSparse [19] as the backbone of our 3D sparse convolution. Drawing inspiration from SparseNeuS [12], we enhance each 3D voxel feature by concatenating the 2D features and RGB colors from the projected 2D pixels across all six views with the voxel's feature in the final attention layer, which is designed to improve the accuracy of color prediction for each voxel. Instead of directly predicting the color of each voxel, here the UNet is tasked with predicting a set of linear weights for each voxel. These weights are subsequently employed to interpolate the colors of the 2D projected pixels. The UNet model is trained with a batch size of 120 and a learning rate of 1e-4. For constructing the ground truth color volume, we start by unprojecting the multi-view images into a 3D space, resulting in a colored point cloud. Each voxel's ground truth color is then determined by interpolating the colors of its nearest neighbors in the fused point cloud.

4. Qualitative Comparison of Multi-View Generation Methods

In Figure 8, we present a qualitative comparison of various multi-view image generation approaches. Techniques like Zero123 [10] and Zero123-XL [3], which do not model the joint distribution of multi-view images, struggle with maintaining 3D consistency across the generated images. This is evident in the inconsistencies observed in the cat tails and the chair. Furthermore, Zero123 [10] tends to produce darker images in certain views, likely due to biases in its training dataset (as seen in the car example). In contrast, concurrent methods such as SyncDreamer [11] and Wonder3D [13] demonstrate improved 3D consistency. However, they sometimes miss capturing fine-grained details or struggle with complex image scenarios. Our approach excels in generating multi-view images that are not only consistent in 3D but also remarkably adept at preserving the intricate details of the input image.

5. Qualitative Examples of Texture Refinement

In Figure 9, we present comparative results showcasing the effect of texture refinement. Although the texture quality of the outcomes produced by the 3D diffusion model can be constrained by the volume resolution, it is possible to further augment the texture quality through a lightweight optimization process. This enhancement leverages the generated 3D consistent multi-view images as supervision.

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[2] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023. 1

[3] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023. 5

[4] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 5

[5] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 1

[6] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. *arXiv preprint arXiv:2303.09556*, 2023. 5

[7] 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. 1

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[9] Lambda Labs. Stable diffusion image variations. https://huggingface.co/lambdalabs/sd-image-variations-diffusers, 2022. 5

[10] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023. 2, 5

[11] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 5

[12] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *European Conference on Computer Vision*, pages 210–227. Springer, 2022. 5

[13] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion, 2023. 5

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[15] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1

[16] 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

[17] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 2

[18] Tim Speed. Flexdiffuse: An adaptation of stable diffusion with image guidance. https://github.com/tim-speed/flexdiffuse, 2022. 3

[19] Haotian Tang, Shang Yang, Zhijian Liu, Ke Hong, Zhongming Yu, Xiuyu Li, Guohao Dai, Yu Wang, and Song Han. Torchsparse++: Efficient training and inference framework for sparse convolution on gpus. In *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2023. 5

[20] Lyumin Zhang. Reference-only control. https://github.com/Mikubill/sd-webui-controlnet/discussions/1236, 2023. 3