Realiz3D: 3D Generation Made Photorealistic via Domain-Aware Learning

Supplementary Material

Appendix

Contents

A. Feature Maps in Diffusion Models

A.1. Motivation

As thoroughly discussed in the *Diffusion Models and Domain Gaps* section (Section 3) of the main paper, prior works has shown that visual details evolve throughout the denoising process in diffusion models, both across timesteps and across layers of the neural network. Building on studies that analyze diffusion features [5, 13], we examine and visualize the internal representations of our diffusion transformer backbone over the course of denoising.

To do so, we first generate diverse prompts describing random objects in random styles (e.g., photorealistic, cartoon, watercolor, anime, comic, etc.) using an LLM. We then synthesize images from these prompts using our base DiT text-to-image model, running generation with 20 DDIM steps.

During generation, we extract intermediate feature maps at multiple timesteps and layers of the network. We apply PCA to these features and visualize the resulting components as RGB images, presented in Fig. 1. Importantly, PCA is computed independently for each timestep–layer pair, so the colors are not consistent across timesteps or layers. Concretely, we extract features at four timesteps (800, 700, 500, and 200, with $T = 1000$) and at three layers corresponding to the beginning, middle, and end of the model. Let N_B denote the total number of DiT blocks.

We present the PCA-reduced features in Fig. 1. The figure comprises four vertically stacked subfigures, where each subfigure shows features from different layers at a **fixed** timestep. Within each subfigure, rows follow the order of the network depth. The top row shows features from an early layer in the architecture, and the bottom row shows features from a later layer. Similarly, the ordering of the subfigures follows the denoising schedule: the top subfigure corresponds to an early timestep at sampling (where the noise level is highest), and the bottom subfigure corresponds to a late timestep (where the noise level is lowest).

As shown in the figure, when the noise level is high, the features primarily capture coarse and structural information, which is largely domain-agnostic. As the noise level decreases, the features increasingly reflect high-frequency patterns and fine-grained details.

The denoising layers exhibit a similar trend: early layers encode coarse structural patterns, shared between synthetic and real data, whereas later layers capture fine details.

A.2. Case Study: Layer-Selective Training in 2D Image Generation

As discussed above and on the main paper, early layers in the diffusion transformer network usually correspond to structural and coarse patterns, while later laters capture fine-details and high-frequency patterns. We leverage this observation in our proposed Layer-Aware Training, where real data affect later layer more strongly and synthetic data has a stronger influence of the early layers.

Before developing our Layer-Aware Training strategy, we conducted a proof-of-concept experiment for image generation, to test whether this insight could be used during training to gain controllability from synthetic data while preserving realism from real data. The results are shown in Fig. 2.

We conducted the proof-of-concept experiment using two equal-sized datasets: one synthetic and one realistic. The synthetic images were rendered from 3D assets, while the realistic images were generated by a T2I model using prompts describing similar objects, augmented with the phrases “highly realistic” and “white background” appended to each prompt.

We fine-tuned the base DiT T2I model in two ways. First, we fine-tuned it solely on synthetic data. Second,

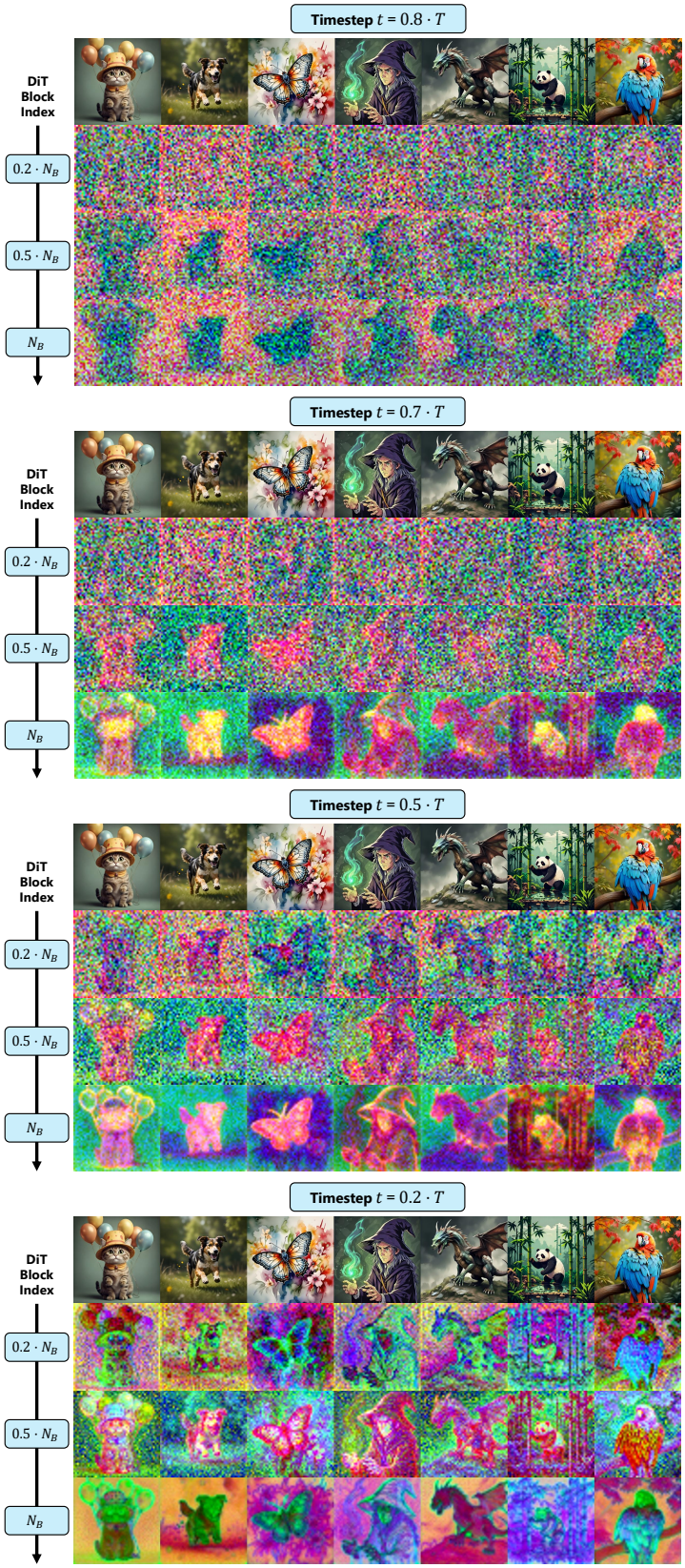

Figure 1. Diffusion features, extracted from our base T2I model at different timesteps and layers during the generation process. We visualize the first three PCA components as RGB images. See Sec. A.1 for additional details and analysis.

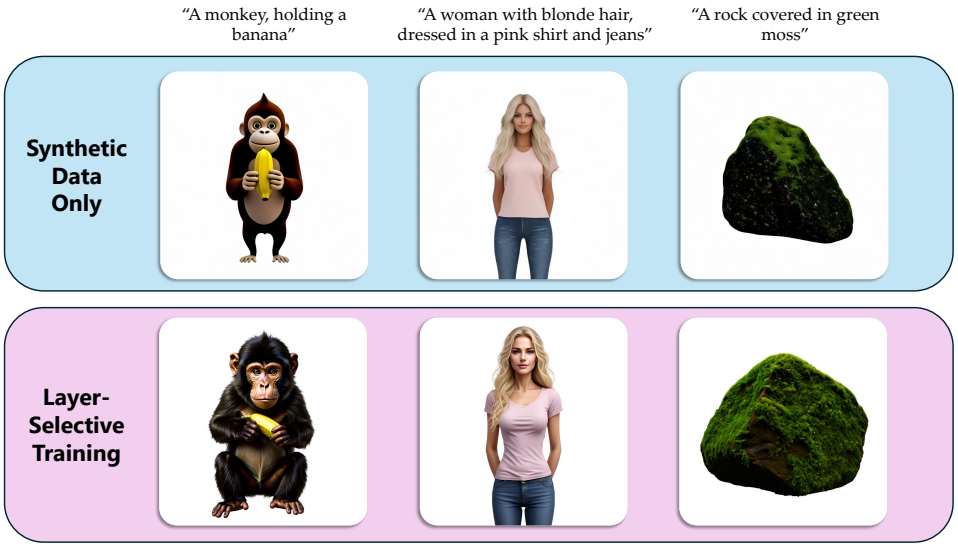

Figure 2. **Layer-Selective Training.** Training early blocks on synthetic data and later blocks on real data enables the model to learn controllable properties from synthetic data while maintaining photorealism. See further details in Sec. A.2.

we applied a layer-selective strategy: the first 50% of DiT blocks were fine-tuned on synthetic data (while the later blocks were frozen), and the remaining 50% were fine-tuned on realistic data (while the early blocks were frozen). Although both approaches caused the model to shift toward object-centric generation with white backgrounds, the layer-Selective strategy consistently preserves the base model's realistic prior more effectively than the full-synthetic baseline, as shown in Fig. 2. While a similar experiment was done with timestep-selective training, we found the layer-selective approach to be more stable and robust.

This proof-of-concept experiment provides the core motivation and inspiration for the layer-aware training approach described in the main paper.

B. Method

In this section, we provide additional information and implementation details about the different components of Realiz3D.

B.1. Domain Shifters (Stage 1)

Discussion: Domain Shifter Design Our design builds on the intuition that our goal is to rebalance existing visual modes within the pretrained model rather than introduce new modalities. Prior adapter-based methods [3, 8] handle much larger modality shifts: AnimateDiff [3] suppresses the low-resolution and noisy video latent pathway to inject temporal conditioning, while Wonder3D adds conditioning streams for normal maps that differ substantially from natural images. In our case, both real and synthetic imagery already occupy the model's learned feature space, making a lightweight low-rank residual sufficient to adjust their balance without disrupting pretrained representations [11].

B.2. Fine-tuning with Representation Binding (Stage 2)

As described in the main paper, we incorporate real samples during training to prevent forgetting realism. Since real images lack explicit control supervision, naïvely training on them could disrupt control learned from synthetic data. To avoid this, we use real data to mostly update the later diffusion blocks, responsible for appearance refinement, while freezing the early blocks.

Concretely, During each real-data training iteration, we freeze DiT blocks $B \in [0, B_i]$, where i is an integer block index randomly drawn from $[0, \tau_B]$. This stochastic layer-freezing regularizes more strongly early representations, without requiring a fixed cutoff. We find the stochastic approach to be robust to small variations and selection of configuration. Empirically, setting $\tau_B \in [0.4, 0.5]$ of the total number of blocks provides stable and robust performance.

Throughout the entire training process, we rely solely on the traditional diffusion loss, which is well established and empirically stable. At stage 1, the diffusion objective is used to train our Domain Shifters, and at stage 2 the same objective is used to train the DiT backbone. When training on synthetic samples, the training objective is $\mathcal{L} = \mathbb{E}_{z,\epsilon\sim\mathcal{N}(0,1),\,t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c, e_{syn})\|_2^2\right]$ (at stage 1, $c = \varnothing$).

When training on realistic samples, the training objective is $\mathcal{L} = \mathbb{E}_{z,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c = \varnothing, e_{real})\|_2^2\right]$.

B.3. Inference-time Domain Shifting

As discussed in the main paper, to further promote control transfer, we introduce *Domain Reassignment*. With probability p_B, we reassign early DiT blocks ($B \in [0, B_j]$, j is an integer, sampled from $[0, \tau_B]$) to operate in synthetic mode even when processing real samples; that is, we substitute $e_{\text{domain}} \leftarrow e_{\text{syn}}$ in the corresponding Domain Shifters.
In practice, we set $e_{\text{domain}} = e_{\text{syn}}$ when either of the following conditions hold:

- *Layer-based shifting:* For all timesteps t and DiT blocks $B \leq B_{\text{max}}$, corresponding to coarse sample structure. We typically set B_{max} to 20%-30% of the total number of blocks N_B.
- *Time-based shifting:* For all DiT blocks and timesteps $t \geq t_{\text{max}}$, corresponding to high noise levels. We find $t_{\text{max}} \in [800, 1000]$ to be most effective.

By tuning B_{max} and t_{max} on a small validation set, users can trade off between stronger control adherence and higher photorealism without any retraining. In our experiments, we prioritize realism, and perform inference-time domain shifting to improve controllability while keeping the realism loss minimal.

We observe that time-based shifting tends to have a stronger effect on realism while layer-based shifting tend to have a mild and stable effect. Therefore, to select the mentioned hyperparameters, we first fix B_{max} by evaluating a small validation set across a few candidate values, ranging from 0% to 40% of the total blocks in 10% increments. Then, we select t_{max} to achieve the desired balance, testing values between 800 and 1000 in steps of 50. For the selection process, we evaluate realism-based metrics and define a threshold that represents the maximum reduction in realism we are willing to tolerate. Since we do not perform an exhaustive search, the entire tuning process takes less than an hour and is performed only once.

C. Implementation Details

C.1. Data

Realistic Training Data. As mentioned in the main paper, to create the realistic dataset, we use the same prompts from the synthetic dataset and generate $V = 4$ photorealistic images using the base T2I model. This ensure fair evaluation, as both datasets contain the same diversity of objects.

As the base model is biased towards the frontal view, we generate each prompt using four viewpoint descriptions ("front", "back", "right side", "left side"), thus collecting $V = 4$ photorealistic images per prompt. The model often fails to follow the desired viewpoint, yet this strategy helps to enrich the data and increase pose variation.

These view descriptions are **not** used during training. During training, the prompts associated to the images do not contain this information, and the random placement of realistic images in the 2×2 grid is not affected by this information.

ImageNet Curation Process for Real-World Realism Metrics While the prior preservation metrics indicate whether the generated images seem realistic, these metrics are based on synthesized images. We aim to ground our evaluation in real-world image data which, to the best of our knowledge, was not used for training our base model nor the pretrained models.

To that end, we processed the prompts describing the evaluation objects through LLM, to find the most fitting categories in the ImageNet dataset, resulting in 42 classes. Then, we use CLIP [10] score to select 4 images per class from the validation set, and segment them using an internal tool similar to Rembg [2].

C.2. Training and Sampling

Training. All evaluated models are trained for 10 epochs on 64 NVIDIA H100 GPUs, using batch size of 8 and learning rate of $5e^{-}5$. Adapters that are trained separately are trained for 3 epochs. We verified that all models and adapters converged and did not reach overfitting. Importantly, throughout the training process we rely only on the standard reconstruction diffusion loss, which is well-established and empirically stable. To prevent adapter-based methods from overfitting to white-background, object-centric layouts, we perform a very short warmup on realistic samples in all experiments that incorporate real data during training.
Sampling. For sampling we use DDIM [12] with 50 steps. Following the base model, we use CFG [4] only for the text condition.
Hyperparameters. For all experiments we set $r = 8$ for Domain Shifter modules.
For Representation Binding, we set τ_B to be 40% of the total number of blocks and $p_b = 0.1$.
For Inference-time Domain Shifting we set B_{max} to be 30% of the total number of blocks and $t_{\text{max}} = 950$. The selection process of B_{max} and t_{max} happens at test-time using a small validation set of 20 held-out objects.

D. Evaluation

D.1. Implementation Details

SDEdit Baseline. We evaluate the effectiveness of the SDEdit [9] mechanism for multiview applications. Specifically, we take the outputs of the synthetic full fine-tuning baseline, add noise to the images, and then denoise them independently using the base pretrained T2I model. The same noise realization is applied to all images of the same grid, while each image is denoised independently. Following the

original paper, we set the noise injection timestep to 500. At higher timesteps, images are expected to exhibit reduced 3D consistency while potentially improving realism. However, even at $t = 500$, SDEdit yields inferior performance to Realiz3D in both 3D consistency and realism.

Pretrained Models. For Text-to-Multiview Generation, we also evaluate TRELLIS [14], a 3D-native model. We use the official text-to-3D "TRELLIS-text-large" model with the original hyperparameters. TRELLIS directly generates a 3D asset and not multiview images, therefore we generate a 3D asset from text and render the 4 orthogonal views (front, back and sides views) that we show across the paper.

Qualitative Results. For all baselines reported with two rank settings (32 and 128), all results shown in the main paper and the supplementary materials were obtained using rank 32.

D.2. Ablation Study: Qualitative Results

To further support the ablation study from the main paper, we present visual examples in Fig. 3 that illustrate the effects of our Representation Binding and Inference-Time Domain Shifting on the model's final outputs. Both techniques improve control adherence while preserving strong realism.

D.3. Additional Qualitative Results

We present qualitative results using the evaluation data described in the main paper, as well as additional test objects from our internal dataset that were held out for evaluation and not used during training.

Multiview Texturing. We present additional qualitative results in Fig. 4, Fig. 5 and Fig. 6, showing all baselines described in the main paper. The corresponding prompt appears in the caption. Although our method uses both normal and position maps as geometric conditions, only the normal maps are shown for readability. Realiz3D achieves significant improvements in photorealism while remaining 3D-consistent and faithful to the geometric conditions.

Text-to-Multiview Generation. We present additional qualitative results in Fig. 7, Fig. 8 and Fig. 9, showing all baselines described in the main paper. The corresponding prompt appears in the caption. Realiz3D achieves notable improvements in photorealism while maintaining strong 3D consistency.

D.4. Text-to-3D Results

We perform text-to-3D generation by backprojecting generated textures onto their corresponding original meshes. Note that we generate only four orthogonal views, which may not fully cover the entire surface.

The results are provided on our project page, where we present side-by-side 3D assets produced by the full-tuning baseline (trained on both real and synthetic data for fairness) and by Realiz3D.

The results highlight that Realiz3D achieves comparable 3D-consistency to the fully synthetic and full fine-tuning baselines, producing coherent and realistic 3D assets.

E. Limitations and Future Work

As mentioned in the main paper, although Realiz3D significantly improves realism, a small gap in control adherence remains. We attribute this to several key factors: (1) 3D consistency is sensitive to fine-grained details. Because the synthetic data primarily contains smooth textures, the task becomes easier for the model. As a result, synthetic baselines tends to produce relatively smooth outputs, reducing the need to learn perfect pixel-level 3D consistency. In other words, these baselines often appear consistent even without having learned accurate pixel-level 3D-consistency. In contrast, Realiz3D produces fine-grained details (e.g., complex textures and materials, hair and fur) that are much more sensitive to even slight misalignment. (2) Domain gaps can occasionally manifest in unrealistic geometry, not just appearance, causing Realiz3D to slightly deviate from the original geometry. (3) The base model's lighting bias can lead to inconsistent appearance. Our synthetic data is uniformly lit, and the generated views largely preserve this property, appearing evenly illuminated. However, the base T2I model shows a bias in lighting for certain objects and materials. An example failure case appears in Fig. 10, where we generate the texture of a hamburger. The hamburger is consistently lit more strongly from the front right, while the back left remains noticeably darker. The same trend is consistent across different seeds. After examining hamburger images online and inspecting those generated by our base T2I model, we find that this lighting condition is extremely common in real-world images, creating a strong lighting bias in the base model. Recent advances in relighting [1, 6, 7] and specifically the use of two synthetic domains, one uniformly lit and one randomly lit, offer promising avenues for addressing this gap.

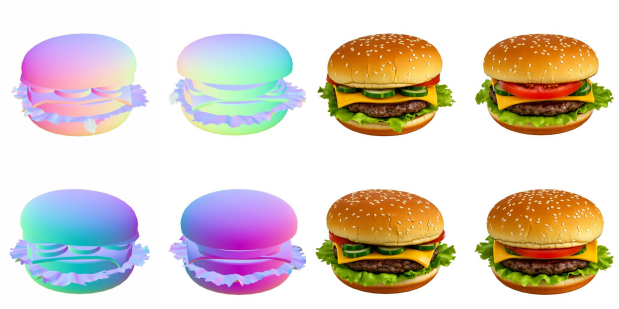

Figure 10. **Multiview Texturing.** Inconsistent lighting caused by the base T2I model's lighting bias.

Figure 3. **Ablation Study.** We demonstrate the importance of our Representation Binding and Inference-time Domain Shifting on Multi-view Texturing. Red circles highlight inconsistent regions with the geometry. Both techniques enhance control adherence, while maintaining realism. The presented prompts: ”A baby stroller with a leather seat and a black plastic container on the bottom”, ”A structure with a flat top, made of natural stone”.

In addition, a natural extension of Realiz3D is to apply our techniques to video diffusion models, which have recently demonstrated remarkable capabilities. When these models are trained to incorporate a 3D condition, they are often fine-tuned on synthetic data, introducing a similar domain gap.

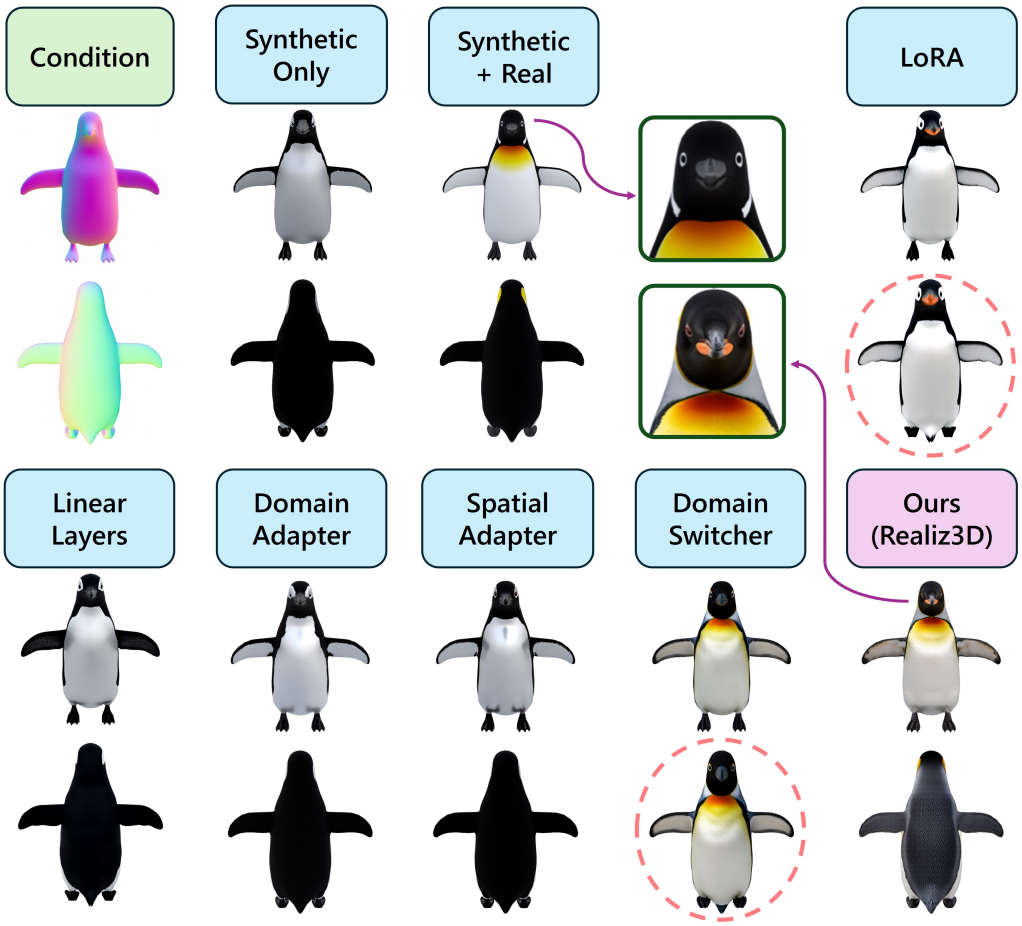

Figure 4. **Multiview Texturing.** "A king penguin, highly realistic and detailed". Red circles highlight inconsistent regions (either with the geometry or with other views). Best viewed zoomed in.

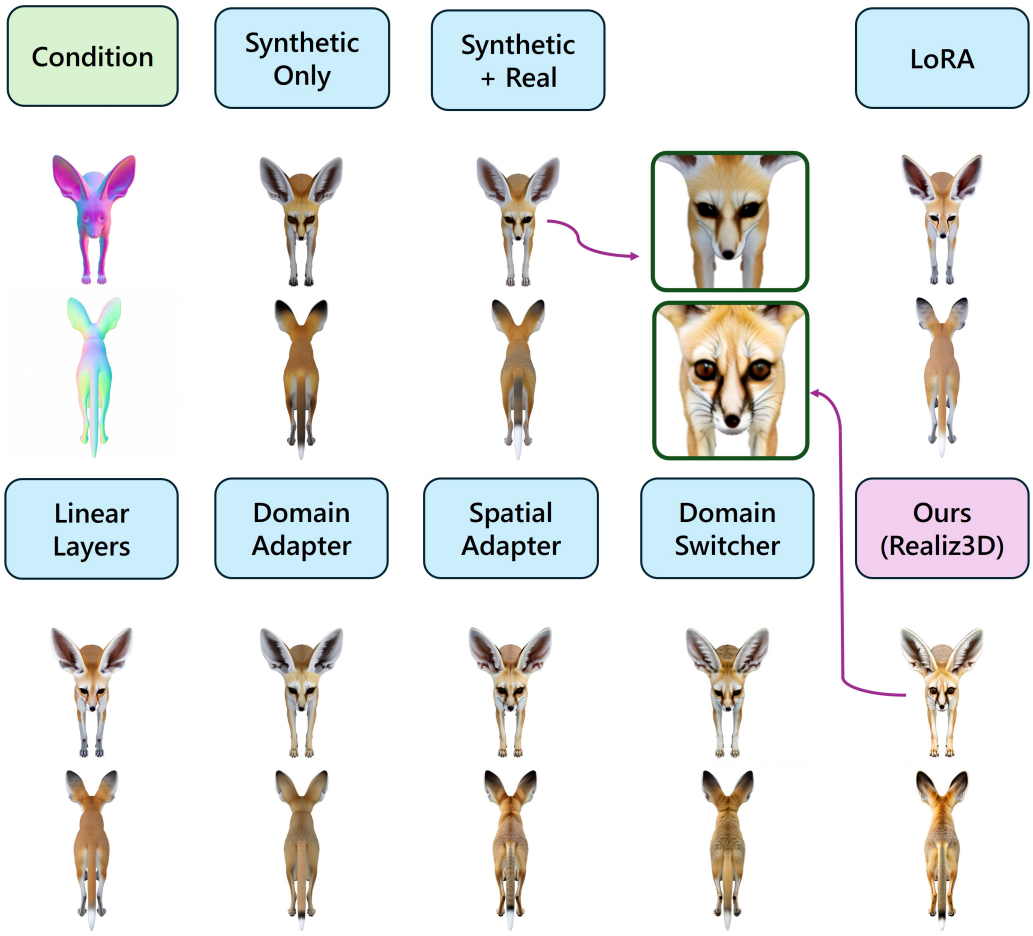

Figure 5. **Multiview Texturing.** ”A fennec fox, highly realistic and detailed”. Best viewed zoomed in.

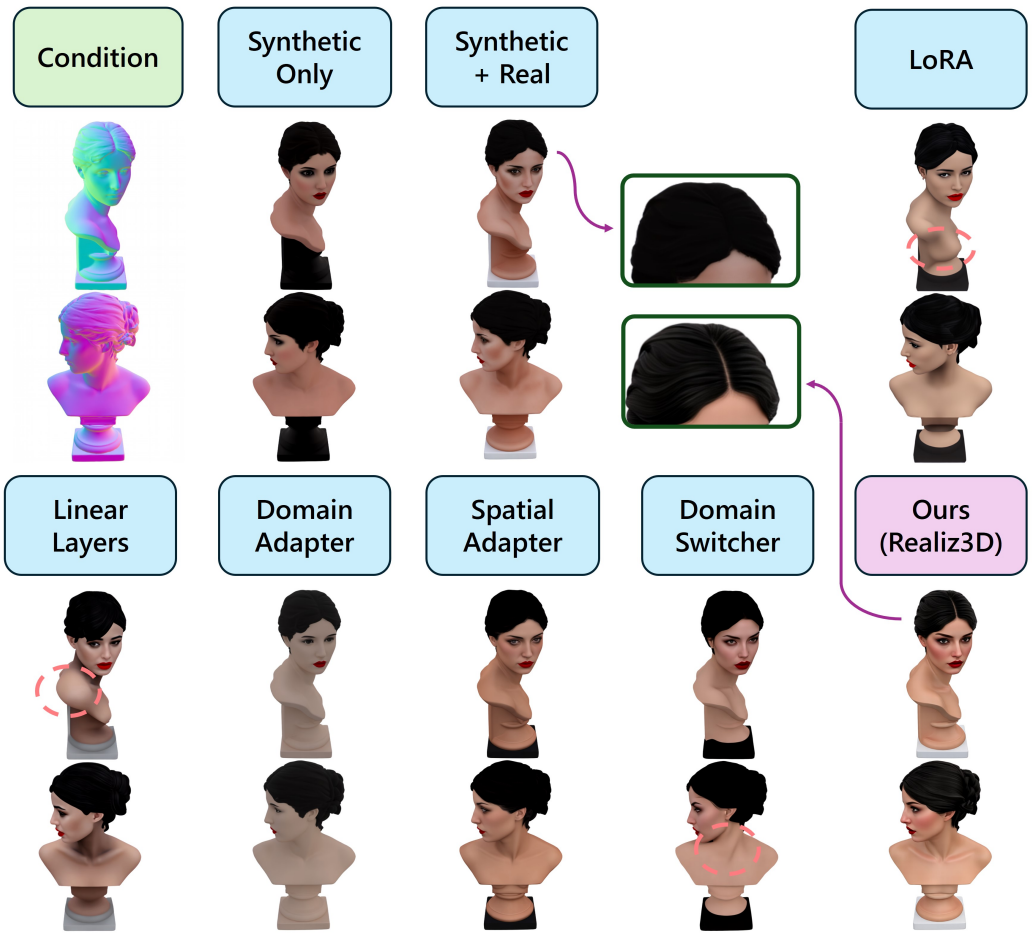

Figure 6. **Multiview Texturing.** "A woman with a dark hair and red lips, highly realistic and detailed". Red circles highlight inconsistent regions (either with the geometry or with other views). Best viewed zoomed in.

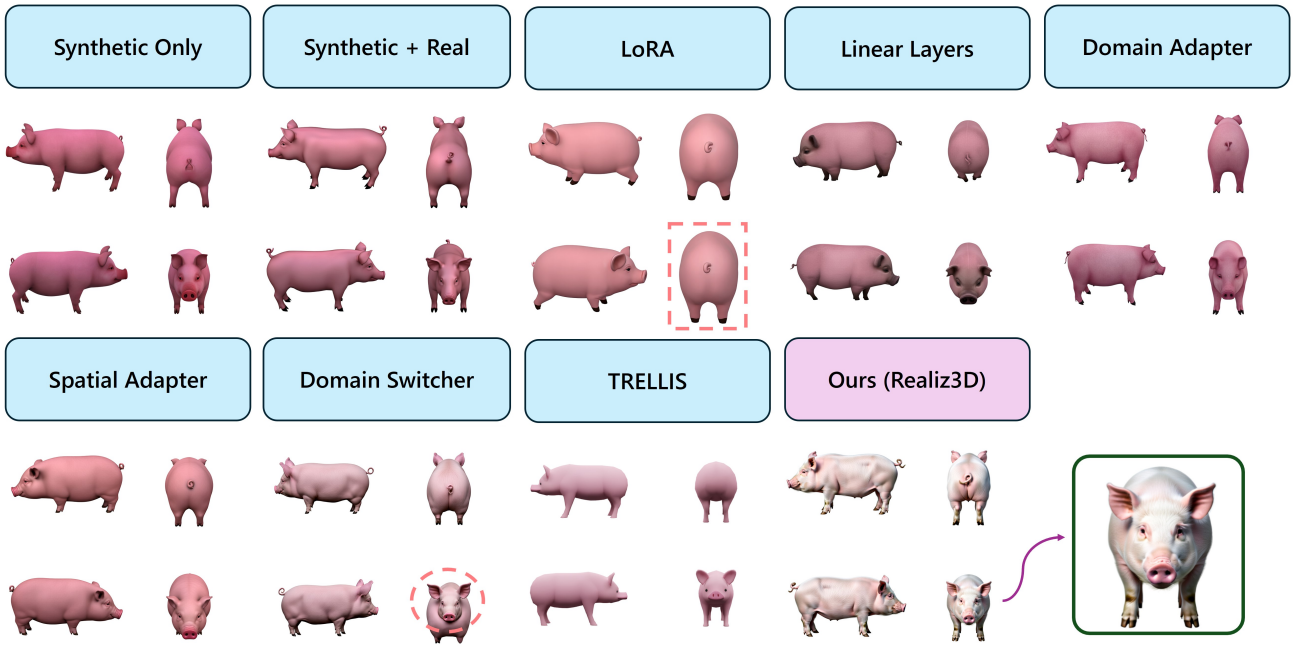

Figure 7. **Text-to-Multiview Generation.** "A pink farm pig, highly realistic and detailed". Red circles/squares highlight inconsistent regions/incorrect viewpoints, respectively. Best viewed zoomed in.

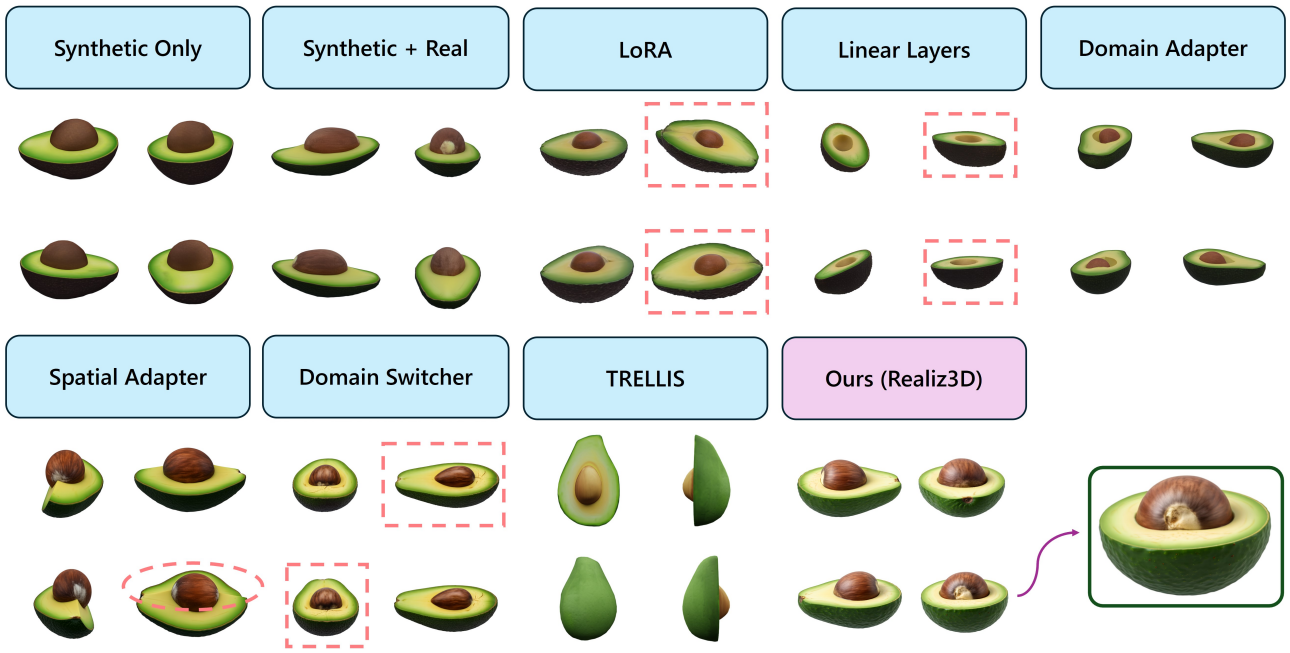

Figure 8. **Text-to-Multiview Generation.** "A half slice of avocado, highly realistic and detailed". Red circles/squares highlight inconsistent regions/incorrect viewpoints, respectively. Best viewed zoomed in.

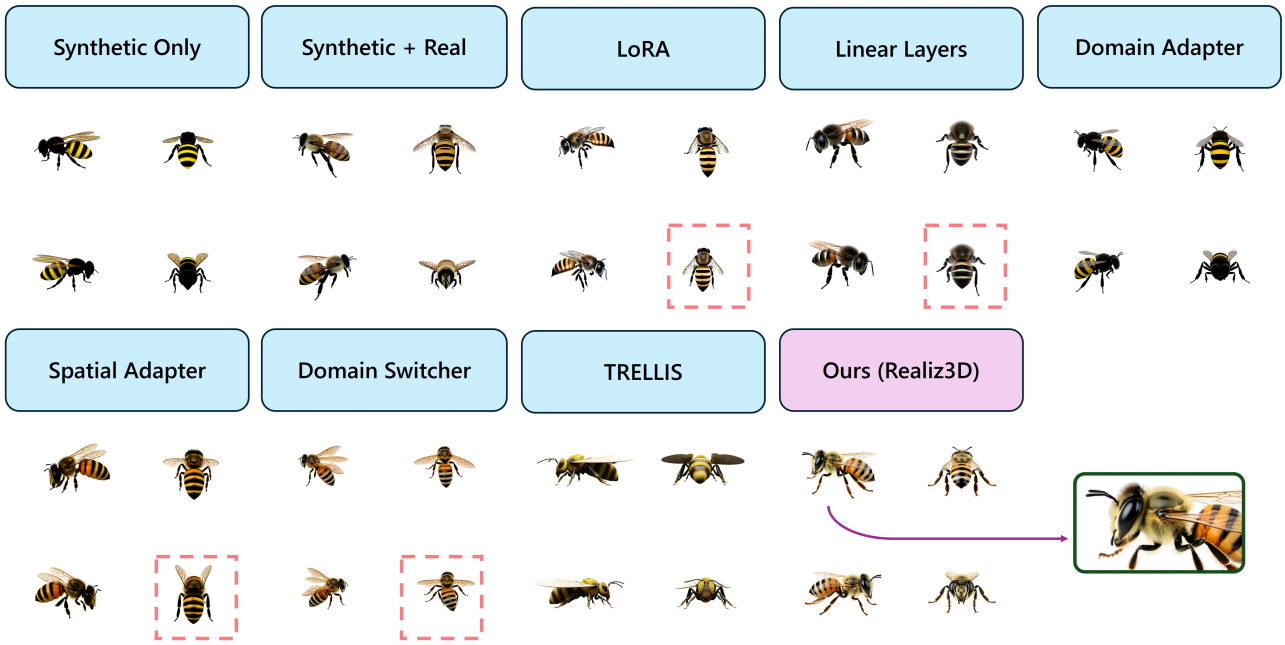

Figure 9. **Text-to-Multiview Generation.** ”A bee, highly realistic and detailed”. Red circles/squares highlight inconsistent regions/incorrect viewpoints, respectively. Best viewed zoomed in.

References

[1] Sumit Chaturvedi, Mengwei Ren, Yannick Hold-Geoffroy, Jingyuan Liu, Julie Dorsey, and Zhixin Shu. Synthlight: Portrait relighting with diffusion model by learning to re-render synthetic faces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 369–379, 2025. 5

[2] Daniel Gatis. Rembg: A tool to remove image backgrounds. https://github.com/danielgatis/rembg, 2025. Accessed: 2025-10-15. 4

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

[4] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

[5] Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025. 1

[6] Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Chih-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, et al. Diffusion renderer: Neural inverse and forward rendering with video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26069–26080, 2025. 5

[7] Yehonathan Litman, Fernando De la Torre, and Shubham Tulsiani. Lightswitch: Multi-view relighting with material-guided diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27750–27759, 2025. 5

[8] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9970–9980, 2024. 3

[9] 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

[10] 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. 4

[11] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *NeurIPS*, 2017. 3

[12] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 4

[13] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023. 1

[14] 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. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21469–21480, 2024. 5