# ViLearn: Accelerating Training Convergence of Image-to-3D Generation via Visibility Learning

## Supplementary Material

In the appendix, we present more implementation details in Sec. A and analyze the effect of visible token fraction on generated geometry during inference in Sec. B. We provide additional visual results of the scaling experiments in Sec. C. Early experiments are reported in Sec. D. We then discuss the limitations of our method in Sec. E. Finally, in Sec. F, we analyze the orthogonal relationship between the two shape representations and outline potential extensions of ViLearn as future work.

## A. More implementation details.

We present the training configuration in Sec. A.1. We then detail the four adopted metrics in Sec. A.2.

### A.1. Training configuration

We train all models using 32 A800 GPUs. For 3D shape VAE, we adopt the pretrained VAE from Dora [1]. This VAE builds upon the architecture of 3DShape2VecSet [23], improving its reconstruction performance through sharp edge sampling, and we use it directly without finetuning. For MM-DiT [5], we apply ViLearn only to the double-stream blocks where image condition is injected, while the shape-only single-stream blocks do not use ViLearn, in order to encourage free learning of internal relationships among shape tokens. We validate the contribution of ViLearn under two configurations. For ablation studies (Sec. 5.3), we consistently employ a small model (0.39B parameters) trained on 0.27M data samples filtered from Objaverse [4]. To demonstrate scalability (Sec. 5.4), all scaling experiments (ViLearn and vanilla training baseline) use a large model (1.1B parameters) trained on an expanded dataset of 1.1M data samples. Tab. S1 details the model architectures for these two configurations. Tab. S2 details the progressive training schedule for each configuration, which specifies the token length, batch size, and image resolution across training steps. This schedule is strictly followed for all models trained under the respective configuration.

### A.2. Metrics.

As described in Sec. 5.2, we use four complementary metrics to evaluate geometry quality for single-image-to-3D generation on a 1,100-image test set. Among these, Floaters and IS-AS are newly introduced in this paper, while GP and GD are computed using the official inference code provided by Hi3DEval [24]. We detail these metrics below.

**Image-Shape Alignment Score (IS-AS).** Accurately measuring how well a generated 3D shape aligns with an input image is crucial, yet there is no agreed-upon metric for local alignment. Existing methods [9, 13] typically adopt global semantic metrics [16, 19] by embedding sampled 3D points and the input image or text prompts into a shared semantic space, which makes them insensitive to fine-grained geometric differences. To address this, we introduce IS-AS, a metric for fine-grained image–shape alignment. Given an input RGB image, we first estimate its normal map using Moge-2 [15]. We then render [7] the generated mesh from 512 uniformly sampled viewpoints on the viewing sphere to obtain 512 normal maps, using nvdiffrast [7] at a resolution of $518 \times 518$ in OpenGL camera coordinates with a camera distance of 3.2 and a $45^\circ$ field of view. For both the estimated and rendered normal maps, we rescale the foreground so that it occupies $90\%$ of the image area, ensuring a consistent silhouette scale. These normal maps are encoded with DINOv2-large [12], and we compute the average local cosine similarity over corresponding patch tokens between the estimated normal map and each rendered one. The final IS-AS is defined as the maximum similarity across all views (Top-1), and Fig. S1 visualizes some cases of the representative Top-5 most similar views for ViLearn.

**Floaters.** We use trimesh [3] to calculate the number of disconnected components (Floaters) of each generated mesh. This metric evaluates geometric integrity directly in 3D space. It provides a comprehensive assessment by analyz-

Table S1. **Model architectures.** $n_{\text{params}}$ denotes the number of parameters; $n_{\text{double}}$ denotes the number of double stream block layers; $n_{\text{single}}$ denotes the number of single stream block layers; $d_{\text{model}}$ denotes the hidden dimension of the model; $n_{\text{heads}}$ denotes the number of heads; $d_{\text{head}}$ denotes the hidden dimension of the heads

| Model size | $n_{\text{params}}$ | $n_{\text{double}}$ | $n_{\text{single}}$ | $d_{\text{model}}$ | $n_{\text{heads}}$ | $d_{\text{head}}$ |
|---|---|---|---|---|---|---|
| Small | 0.39B | 5 | 4 | 1536 | 16 | 96 |
| Large | 1.1B | 10 | 20 | 1536 | 16 | 96 |

Table S2. **Training schedule.** We progressively increase the token length and adjust the batch size (per GPU) during training. The schedule differs for the ablation (Small) and scaling (Large) experiments.

| Model Size | Training Steps | Token Length | Batch Size | Image Resolution |
|---|---|---|---|---|
| Small | 0K – 20K | 256 | 512 | 256 |
| | 20K – 40K | 1024 | 256 | 256 |
| | 40K – 110K | 2048 | 208 | 256 |
| Large | 0K – 20K | 256 | 512 | 256 |
| | 20K – 40K | 1024 | 256 | 256 |
| | 40K – 80K | 2048 | 152 | 256 |
| | 80K – 110K | 4096 | 76 | 518 |

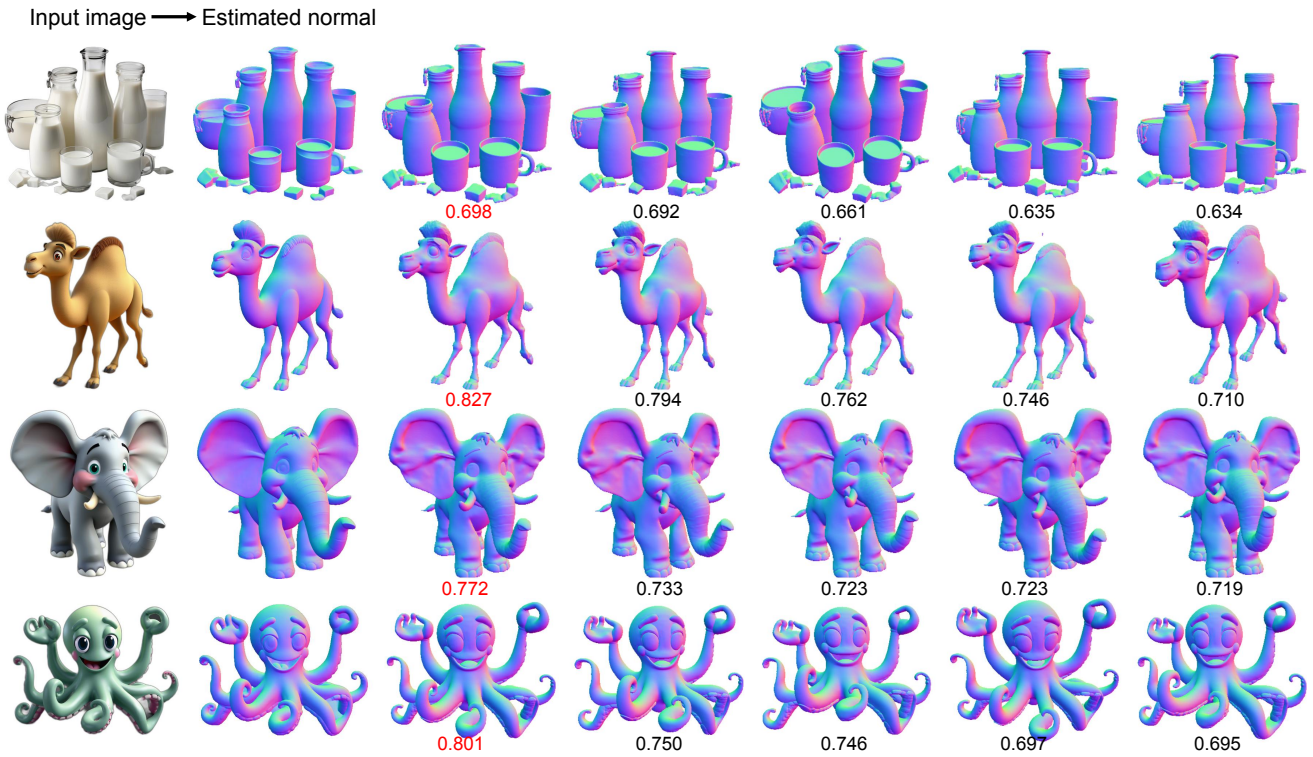


Figure S1. **Top-5 similarity scores between the estimated normal map and the rendered normal map.** We define the highest score as the Image-Shape Alignment Score (IS-AS) and highlight it in red.

ing not only the visible surfaces but also the integrity of internal, occluded geometry. Fewer floaters signify a more structurally sound and complete mesh, thus reflecting better geometric quality.

**Geometry Plausibility (GP) & Geometry details (GD).** These two metrics evaluate the plausibility and detail of visible geometry from multi-view renderings. Hi3DEval [24] renders meshes generated by different methods from multiple viewpoints, collects human and GPT-4o preference scores, and trains a scoring model based on these annotations. GP and GD measure geometric quality purely in 2D without considering alignment between geometry and the input images, so they are complementary to metrics of Floaters and IS-AS. Hi3DEval [24] also provides a Prompt Alignment (PA) metric, while it requires textured multi-view renderings and therefore cannot focus on the alignment between geometry alone and images. We do not use PA for this reason. Moreover, since Hi3DEval [24] scoring model is trained on limited human preference data, its generalization and fairness might be weaker than those of large-scale vision encoders such as DINOv2 [12], and we therefore report GP and GD only as reference metrics.

## B. Effect of visible shape token fraction

In Sec. 4.3, we analyze the distribution of visible shape token lengths in the training set and observe that it peaks when the visible part is roughly one-third of the total token length, then decays towards both sides. This suggests that, on average, the geometry visible from a single view accounts for about one-third of the full 3D geometry in the training data. Motivated by this, at inference time, we take the first one-third of the shape tokens as "visible" and the remaining two-thirds as "invisible", so that our choice of the visible fraction matches the peak of the training statistics.

In this section, we further investigate how the fraction of visible tokens at inference time affects the generated geometry. Using ViLearn-Large (110K checkpoint, 4096 total tokens), we test visible fractions of 1/4, 1/3, and 1/2, as shown in Fig. S2. We observe that a smaller visible fraction encourages the model to generate richer details in occluded regions. As the visible fraction increases to 1/2, the level of detail in the unseen geometry noticeably diminishes. For example, the dragon's tail spikes and the car's interior cabin details (seats, steering wheel) gradually disappear with a larger visible fraction. These results demonstrate that ViLearn allows users to explicitly control the detail level of the generated geometry by simply adjusting the visible fraction of the shape tokens.

## C. More visual results of scaling experiments.

In Sec. 5.4, we validate the scalability of ViLearn qualitatively and quantitatively, and compare it against state-of-

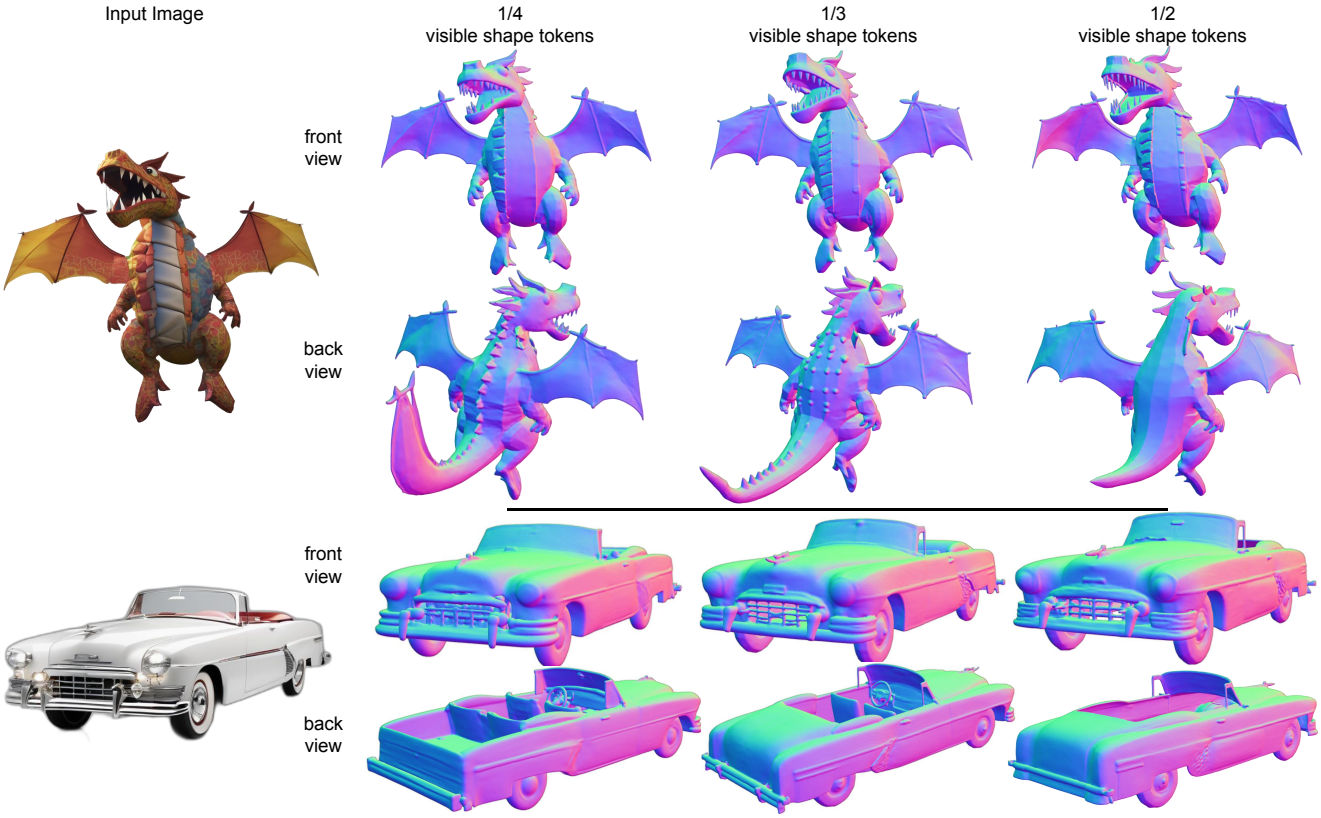


Figure S2. The effect of varying the visible token fraction at inference time. A smaller visible fraction (e.g., 1/4) encourages the model to generate richer details in occluded regions, while a larger fraction (e.g., 1/2) results in less detailed completions.

the-art (SOTA) VecSet-based models [9, 10, 13] of comparable size (1.1B parameters). We note that a strictly fair comparison with SOTA models is challenging, as they are trained with substantially larger and often undisclosed computational and data budgets. For example, TripoSG [10] is trained for 3 weeks across 160 A100 GPUs on 1M data samples, and Step1X-3D [9] uses 96 A800 GPUs for 200K steps on 2M data samples, while Hunyuan3D [13] does not report its GPU count or data scale. In this work, we focuse on training paradigm: we strictly match all training conditions between ViLearn and the vanilla training method (Baseline), ensuring a fair comparison. Under these settings, the vanilla training framework without explicit positional encoding used by the SOTA methods [9, 10, 13] converges more slowly and achieves inferior final performance compared with ViLearn. Benefiting from the acceleration and performance improvements provided by ViLearn, we surpass these SOTA methods [9, 10, 13] while using substantially fewer computational resources. As shown in Fig. S5, we provide more visual comparison of ViLearn with the vanilla baseline and the SOTA models [9, 10, 13]. We exclude the comparison with commercial models, as their training configurations are not clear and their APIs are slow for the inference of the entire 1,100 test set.

## D. Early experiments

In our early experiments, we aim to transfer the success of accelerating the training convergence of image generative models via representation alignment [8, 20, 22, 25], such as VA-VAE [20], to the domain of 3D native generation. Specifically, VA-VAE [20] fine-tunes the pretrained image VAE with an additional regularization loss computed between the structured 2D latent codes and DINOv2 [12] features. However, this strategy cannot be directly applied to a 3D shape VAE. The latent codes of a shape VAE are designed to encode the complete 3D geometry, whereas DINOv2 features are extracted from a single rendered view and thus only capture view-dependent information. As a result, the information carried by each shape token and each image token is not aligned on a per-token level.

To address this mismatch, we introduce an adaptor composed of cross-attention layers. We treat the shape tokens as keys and values, and inject a set of view-dependent queries so that the adaptor output can be aligned with the DINOv2 features of the corresponding view via cosine loss. As illustrated in Fig. S3, we investigate two ways of constructing the queries. The first design uses visible points [14] sampled from the rendered visible surface as queries. The

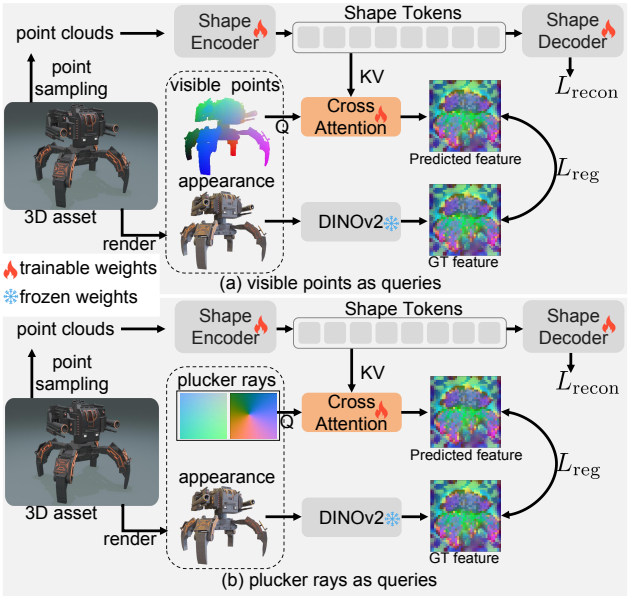


Figure S3. **Two ways to add the regularization loss to shape VAE latent space with DINOv2 feature.** (a) Use the visible points [14] as querires. (b) Use Plucker rays as queries.

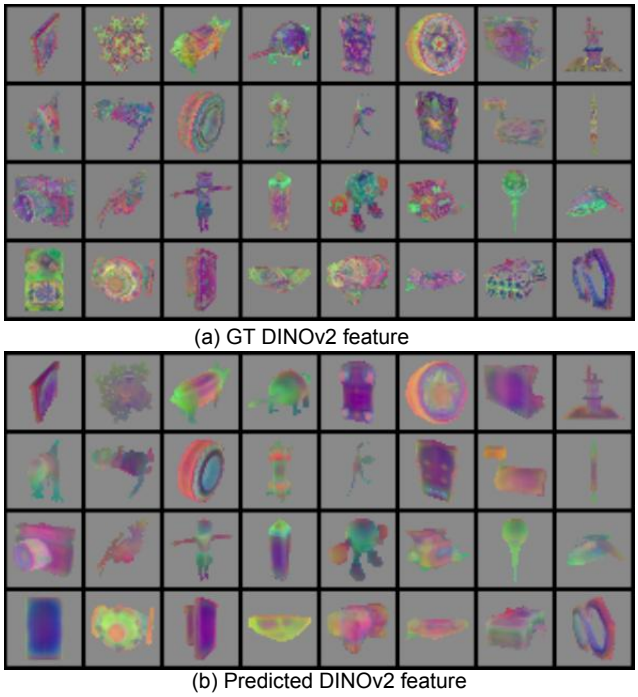


Figure S4. (a) Ground truth (GT) DINOv2 feature. (b) Predicted DINOv2 feature.

second design uses Plücker rays, derived from the camera parameters of the conditioning image used in diffusion training, as queries. Formally, the training objective is defined as $\mathcal{L}=\mathcal{L}_{\text{recon}}+\lambda_{\text{reg}}\mathcal{L}_{\text{reg}}$, where $\mathcal{L}_{\text{recon}}$ denotes the reconstruction loss of the 3D shape VAE, $\mathcal{L}_{\text{reg}}$ denotes the feature-alignment regularization loss between the adaptor outputs and DINOv2 features, and $\lambda_{\text{reg}}$ is a weighting coefficient. We conduct a systematic hyperparameter study for both types of queries, varying the depth of the adaptor (1 or 2 cross-attention layers) and the regularization weight $\lambda_{\text{reg}}\in\{0.1, 0.5, 1.0\}$. After fine-tuning, all these configurations are able to successfully predict the pattern of DINOv2 [12] features for the corresponding views, as illustrated in Fig. S4. However, when we plug these fine-tuned shape VAEs based on Dora [1] into the vanilla training pipeline described in Sec. 3 for image-to-3D diffusion, we observe that the impact on the final 3D generation quality is surprisingly minor. We attribute this to two reasons. First, although representation alignment losses [8, 20, 22, 25] are effective in text-to-image generation and can improve semantic metrics such as FID, they might be less suitable for image-to-3D tasks, which require accurate local image–shape alignment rather than global semantic alignment. Second, the latent features from the image VAE and the DINO features are both structured 2D representations with pixel-level correspondence, so simple MLP layers are sufficient to compute the regularization loss between them. In contrast, the latent features from the shape VAE are 3D representations and do not have a direct correspondence with 2D DINO features. Even when cross-attention is used to project the 3D features to enable loss computation, this approach does not fundamentally resolve or alleviate the ambiguity inherent in single-image-to-3D training. This observation and the above two analyses inspire the hypothesis about the main bottleneck in training image-to-3D models discussed in the introduction (Sec. 1) and motivate our proposed ViLearn (Sec. 4).

## E. Limitation

The limitation of our method lies in the additional computational overhead incurred during the data preprocessing stage. Specifically, visibility grouping (Sec. 4.1) requires rendering a corresponding visible points map for each potential conditioning image used in diffusion model training. Furthermore, it needs the pre-extraction of indices for both visible and invisible tokens from the set of shape tokens. However, this issue can be substantially mitigated through parallelized data processing. In our implementation, we leveraged 64 GPUs to process the entire training dataset of 1.1 million data samples within half a day.

## F. Discussion

We discuss the relationship between VecSet representations and Voxel-based representations. And we discuss the potential extensions of our proposed visibility learning to be explored in future work.

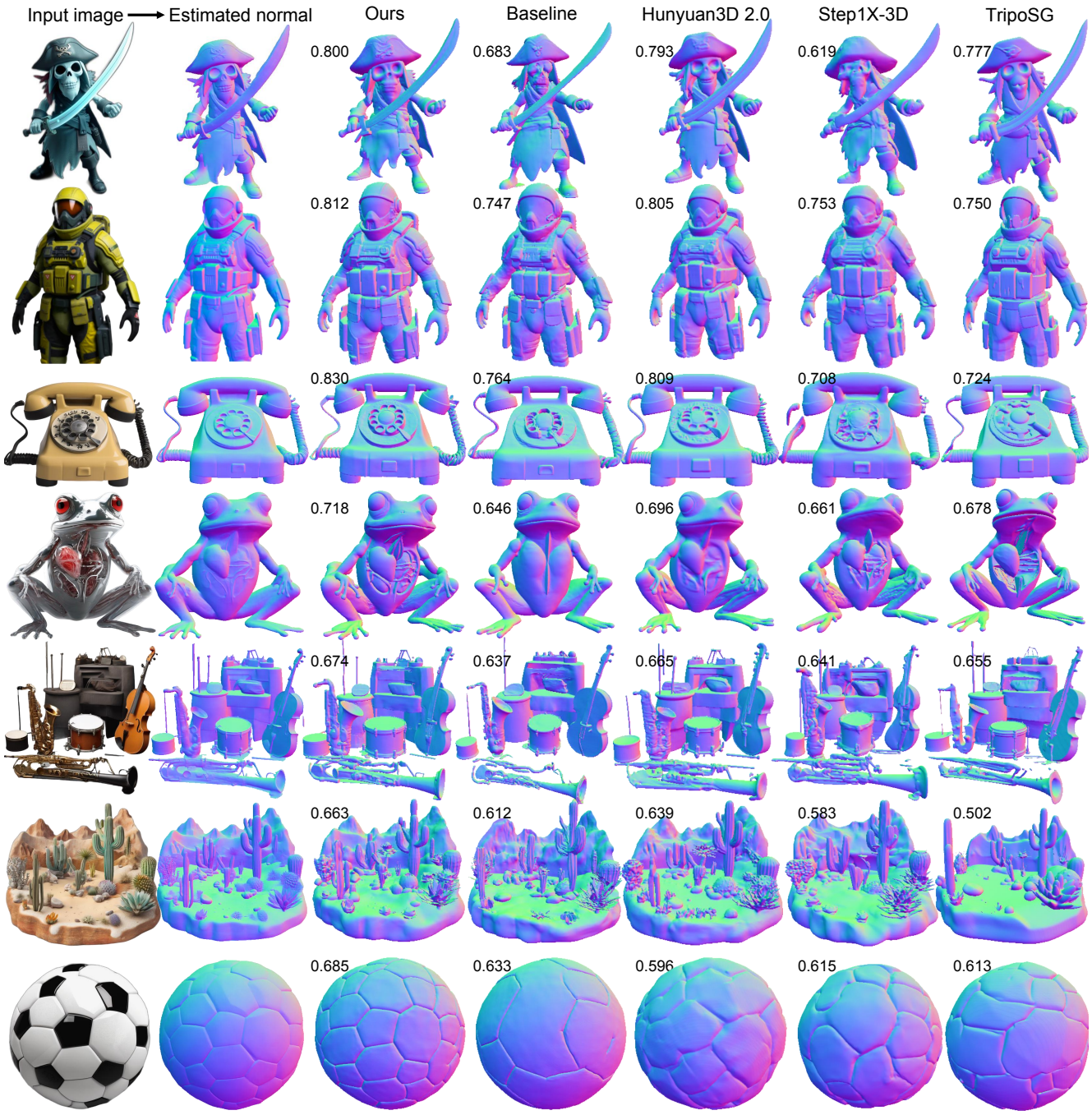


Figure S5. .**Qualitative Comparison: Our Scaled Model vs. Scaled Baseline vs. SOTAs**. For Single-Image-to-3D, we compare the quality of generated meshes among our scaled model (ViLearn), a scaled baseline, and state-of-the-art (SOTA) open-sourced methods, all with comparable model parameters: 1.1B (ours, baseline, Hunyuan3D [13]) and 1.3B (TripoSG [10], Step1X-3D [9]), using the **Image-Shape Alignment Score** (range 0-1, higher is better). Note that while SOTA methods [9, 10, 13] typically train on hundreds of GPUs for several weeks, we achieve the best alignment with only 32 GPUs by accelerating convergence with ViLearn.

## F.1. Relationship of two shape representations.

In this paper, we exclude comparisons with SOTA voxel-based generative models [2, 6, 11, 17, 18, 21], since these two representations are orthogonal. Voxel-based representations achieve the highest reconstruction fidelity but require a large number of shape tokens, leading to staged generation pipelines: voxel locations are first generated as base geometry, and latent features are then predicted conditioned on these locations to refine the geometry. In con-

trast, VecSet representations use compact latent shape tokens, enabling end-to-end and relatively efficient training while maintaining competitive quality.

Despite this difference, the two representations can be combined to leverage their respective strengths. One can first generate a base geometry in the VecSet representation and then feed it into a voxel-based model as a refinement stage to enrich high-frequency geometric details. In such a hybrid pipeline, the primary role of the VecSet-based stage is to produce an image-aligned base geometry with as few floating artifacts as possible; otherwise, misalignment and floaters in the base geometry tend to accumulate in the refinement stage. Consequently, exploring how to accelerate the training convergence of both VecSet-based and voxel-based models is equally important for achieving image-to-3D generation with high geometric detail and strong image alignment at a lower overall training cost. Such a hybrid pipeline has been explored and supported by Ultra3D [2], which uses a VecSet-based base geometry for subsequent Voxel-based refinement.

### F.2. Extensions of ViLearn

**Multi-view conditioning.** For multi-view conditioning, our learnable VA-LPE is more suitable than VA-RoPE as it adaptively weights tokens across views to capture complex relationships, while VA-RoPE relies on fixed positional encodings. However, as observed in Sec. 5.3, VA-LPE exhibits slower convergence due to the absence of VA-RoPE's inductive bias, despite achieving superior final performance. This motivates a hybrid design: augmenting VA-RoPE's rotation matrices with learnable embeddings. Such a formulation combines the benefits of both approaches by providing strong initial bias and adaptive learning capacity, making it applicable to a broader range of tasks.

**Voxel-based representation.** ViLearn can be extended to voxel-based representations [2, 6, 11, 17, 18, 21]. We can extract visible sparse voxels directly from the input image during data preprocessing and apply ViLearn for training. For inference, we could first identify the most similar canonical view via the IS-AS metric and then extract the voxel indices corresponding to this view to acquire the indices of the visible sparse voxels. This allows the extension of VG and VAPE modules to perform visibility learning.

## References


[1] Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 16251–16261, 2025. 1, 4

[2] Yiwen Chen, Zhihao Li, Yikai Wang, Hu Zhang, Qin Li, Chi Zhang, and Guosheng Lin. Ultra3d: Efficient and high-fidelity 3d generation with part attention, 2025. 5, 6

[3] Dawson-Haggerty et al. trimesh, 2025. 1

[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. *arXiv preprint arXiv:2212.08051*, 2022. 1

[5] Patrick Esser, Sumith Kulal, A. Blattmann, Rahim Entezari, Jonas Muller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *ArXiv*, abs/2403.03206, 2024. 1

[6] Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. *arXiv preprint arXiv:2503.21732*, 2025. 5, 6

[7] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. 1

[8] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. *arXiv preprint arXiv:2504.10483*, 2025. 3, 4

[9] Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025. 1, 3, 5

[10] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *CoRR*, abs/2502.06608, 2025. 3, 5

[11] Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *arXiv preprint arXiv:2505.14521*, 2025. 5, 6

[12] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 1, 2, 3, 4

[13] Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025. 1, 3, 5

[14] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 4

[15] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 1

[16] Nina Wiedemann, Sainan Liu, Quentin Leboutet, Katelyn Gao, Benjamin Ummenhofer, Michael Paulitsch, and Kai Yuan. Unifi3d: A study on 3d representations for generation and reconstruction in a common framework. *arXiv preprint arXiv:2509.02474*, 2025. 1

[17] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, and Yao Yao. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *CoRR*, abs/2505.17412, 2025. 5, 6

[18] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 21469–21480. Computer Vision Foundation / IEEE, 2025. 5, 6

[19] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1179–1189, 2023. 1

[20] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 3, 4

[21] Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *arXiv preprint arXiv:2503.22236*, 2025. 5, 6

[22] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025. 3, 4

[23] Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Trans. Graph.*, 42(4):92:1–92:16, 2023. 1

[24] Yuhan Zhang, Long Zhuo, Ziyang Chu, Tong Wu, Zhibing Li, Liang Pan, Dahua Lin, and Ziwei Liu. Hi3deval: Advancing 3d generation evaluation with hierarchical validity, 2025. 1, 2

[25] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders, 2025. 3, 4