PARTIAL3D: CONTROLLABLE 3D ASSET GENERATION FROM MULTIPLE PARTIAL IMAGES

Anonymous authors
Paper under double-blind review

ABSTRACT

While 3D generative models have achieved unprecedented visual quality, controlling these systems remains an underexplored frontier compared to 2D generation, primarily due to the significant data bottleneck associated with high-quality 3D assets. Consequently, existing approaches for 3D control often rely on indirect pipelines that leverage 2D control methods and lift the results to 3D, a process that frequently leads to error accumulation and lacks an efficient feed-forward mechanism. In this paper, we propose Partial3D, a native 3D generative model that utilizes partial images, optional 3D layouts, and text for precise and controllable 3D synthesis, enabling native 3D control without relying on intermediate 2D pipelines. Our main insight is to treat partial images as separable visual conditions, allowing for compositional control over the generated 3D content, analogous to how text prompts are used in 2D generation. To address the inherent ambiguity of composing partial images, Partial3D incorporates optional 3D layout guidance through a Spatial Prompt Modulator (SPM) and text guidance through a training-free velocity-fusion strategy. By treating partial images as separable conditioning signals, while incorporating 3D layouts and text prompts, Partial3D enables native 3D editing through image replacement and supports unified control across image, text, and 3D layout modalities. Extensive experiments demonstrate that Partial3D generates 3D content under precise multimodal control and successfully enables downstream applications such as high-fidelity 3D editing.

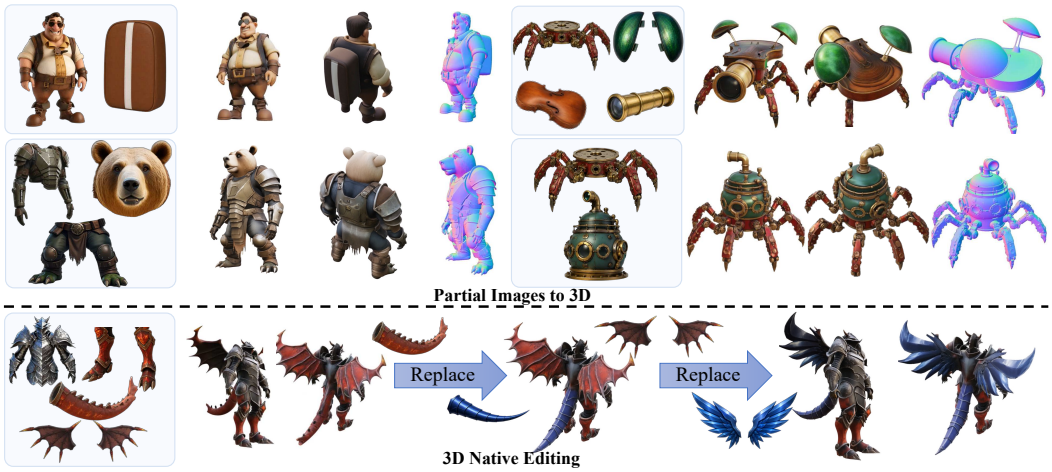

Figure 1: Given multiple partial images, Partial3D generates coherent 3D assets that preserve the visual characteristics of the input images. By treating partial images as separable visual conditions, Partial3D also enables native 3D editing through simple image replacement.

1 INTRODUCTION

While 3D generative models have achieved unprecedented visual quality, controllability remains less explored than in the 2D image domain. In 2D generation, prior works have introduced spatial control through text prompts and bounding boxes Wang et al. (2024); Zhou et al. (2024), incorporated

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

visual identities from reference images Garibi et al. (2025); Chen et al. (2025); Kumari et al. (2023), and further unified image, layout, and text conditions Peng et al. (2025). Furthermore, 2D editing has become remarkably proficient through inversion-based techniques Huberman-Spiegelglas et al. (2024); Rout et al. (2025); Mokady et al. (2022) and flow-based methods Kulikov et al. (2025); Kim et al. (2026), which enable users to manipulate existing images with high precision. However, since these methods are typically built on text-conditioned backbones such as Stable Diffusion Rombach et al. (2021), incorporating image conditions often requires test-time optimization or additional training to align the text and image conditioning spaces Tarrés et al. (2025); Parmar et al. (2025); Dalva et al. (2025). Although image-conditioned 3D generation operates directly on visual conditions and therefore avoids the same text–image domain-gap issue, achieving controllable 3D generation and editing remains challenging as described below.

Most existing 3D editing methods Li et al. (2025); Gat et al. (2026); Hu et al. (2026) still rely on performing 2D edits first and then lifting the edited results to 3D using methods such as SDS Poole et al. (2022) or flow-based distillation Kulikov et al. (2025). As also noted by prior works Ye et al. (2026); Sella et al. (2026); Chi et al. (2026), such indirect 2D-to-3D editing pipelines are prone to error accumulation and often struggle to preserve multi-view consistency and object identity. Furthermore, these lifting-based pipelines are predominantly text-conditioned, making them less visually specific than direct image-based control. Recent feed-forward native 3D works support multimodal or structured controls Ye et al. (2026); Team (2025); Yan et al. (2026), while Fuse3D Jin et al. (2025) further enables multi-image fusion, image-to-3D correspondence, and regional control. However, Fuse3D initializes the global structure from a holistic image, with additional image regions primarily controlling their corresponding local areas. In contrast, Partial3D treats independently cropped partial images, potentially from different source objects, as separate visual conditions and jointly infers their global 3D organization and a coherent complete structure. This formulation also permits component-level composition and editing by adding, removing, or replacing individual partial-image conditions. Moreover, Figure 2 also illustrates that even strong commercial 2D generation models OpenAI (2026) may suffer from multi-view inconsistency when used for such tasks. CAST3D Sun et al. (2026) instead composes 2D assets under text guidance using a training-free two-stage pipeline: it first infers a segmented 3D layout and then generates and integrates components separately. In contrast, our method jointly conditions a native 3D backbone on all partial images and generates the composed asset in a feed-forward manner. This avoids intermediate 3D segmentation and refinement steps.

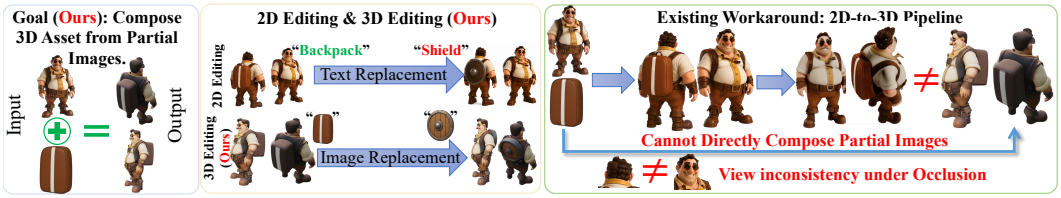

Figure 2: Key motivation of Partial3D.

To tackle these limitations, we propose Partial3D, a 3D generative model for controllable 3D synthesis from partial images. Our key idea is to move beyond the conventional holistic-image-to-3D setting, and instead formulate 3D generation as compositional synthesis from multiple partial-image conditions. To this end, we train Partial3D to compose 3D assets from separate partial images, making visual conditions behave more like compositional units in a text prompt. However, partial-image composition is inherently ambiguous in scale, spatial arrangement, and global structure. To address this, we introduce a Spatial Prompt Modulator for optional 3D layout grounding and a training-free velocity-fusion strategy for text-guided global structure control while preserving fine-grained visual details. By treating partial images as separable compositional units, analogous to words in a text prompt, our framework brings prompt-style replacement to image-conditioned 3D generation and enables native 3D flow-based editing through image replacement. At the same time, it provides a unified generation pipeline that seamlessly integrates partial images, text, and optional 3D layout control within a single native 3D generative framework. Overall, our contributions are as follows:

- To the best of our knowledge, we propose Partial3D, the first feed-forward native 3D framework to jointly synthesize coherent assets from independently cropped partial-image conditions, with optional 3D layout and text controls.

- We introduce a spatial prompt modulator to inject optional 3D layout information, enabling precise spatial control while maintaining the flexibility for layout-free generation.
- We demonstrate a training-free velocity-fusion strategy for optional text guidance by combining image- and text-conditioned models in a shared sparse-structure latent space.

2 Related Work

2.1 Controllable 2D Generation

Recent advances in large-scale text-to-image diffusion models Rombach et al. (2021) have significantly improved the quality and diversity of image generation. However, controlling generation solely through text prompts is often insufficient for precise visual synthesis, motivating a broad range of works that introduce additional conditioning modalities such as layouts, reference images, and personalized concepts. For spatial control, some works enable users to specify object locations or instance-level layouts during generation Wang et al. (2024); Zhou et al. (2024). Although these methods provide effective layout control, their generation process is still primarily driven by text descriptions, which are often less visually specific than direct image-based conditions. MUSE Peng et al. (2025) further combines image, layout, and text conditions in a unified 2D generation framework, making it more closely related to our multimodal control setting. However, since its visual conditions are encoded through CLIP features and used within a 2D image-generation backbone, preserving fine-grained image details remains challenging, and the method does not directly address native 3D generation. Another line of work injects visual concepts or reference identities into text-to-image models Kumari et al. (2023); Garibi et al. (2025); Chen et al. (2025). These approaches improve visual specificity by incorporating image-derived concepts, but typically require per-concept or per-image optimization, which limits their efficiency and scalability. Such limitations largely stem from their reliance on text-conditioned backbones: they either control generation through text, or introduce visual information through additional optimization. For image editing, inversion-based methods Huberman-Spiegelglas et al. (2024); Rout et al. (2025); Mokady et al. (2022) and recent flow-based approaches Kulikov et al. (2025) allow users to edit images by replacing source prompts with target prompts. However, these editing mechanisms also rely on text-conditioned backbones, where controllability is achieved through manipulable linguistic tokens rather than directly composable image conditions.

2.2 3D Generative Models

Image-to-3D generation provides a more visually grounded interface than text-to-3D generation, since the input image directly specifies object appearance, geometry cues, and fine-grained visual identity. Compared with text prompts, image conditions reduce the modality gap between user intent and the target 3D content, making them especially suitable for high-fidelity and identity-preserving 3D synthesis. Early text- or image-guided 3D generation methods often relied on powerful 2D diffusion priors and optimized 3D representations through score distillation sampling or related distillation objectives Poole et al. (2022); Wang et al. (2023). While these optimization-based pipelines greatly advanced 3D generation, they are typically inefficient and may suffer from error accumulation caused by repeatedly distilling 2D priors into 3D representations.

To improve efficiency and 3D consistency, recent works have explored native 3D generative models that directly learn generative priors over 3D representations. These methods introduce various compact and expressive 3D latent spaces, such as structured latent representations Xiang et al. (2025b;a) and vector-set Zhang et al. (2023); Lai et al. (2025), enabling feed-forward generation of 3D content from image conditions. By operating directly in native 3D representation spaces, these models achieve stronger geometric coherence and substantially faster inference than per-instance optimization-based methods. However, existing image-to-3D models usually treat the input image as a holistic conditioning signal, making it difficult to compose multiple partial visual observations or control individual object components independently.

2.3 Controllable 3D Generation

Controlling 3D generative models has been studied from several perspectives, including 3D editing and explicit structural control. Since 2D editing methods controlled by text or other modalities have become highly mature, many early 3D editing methods leverage 2D editing priors and use score distillation sampling to lift the edited results into 3D Haque et al. (2023); Li et al. (2023). However, such optimization-based pipelines are often inefficient and may produce suboptimal 3D results due

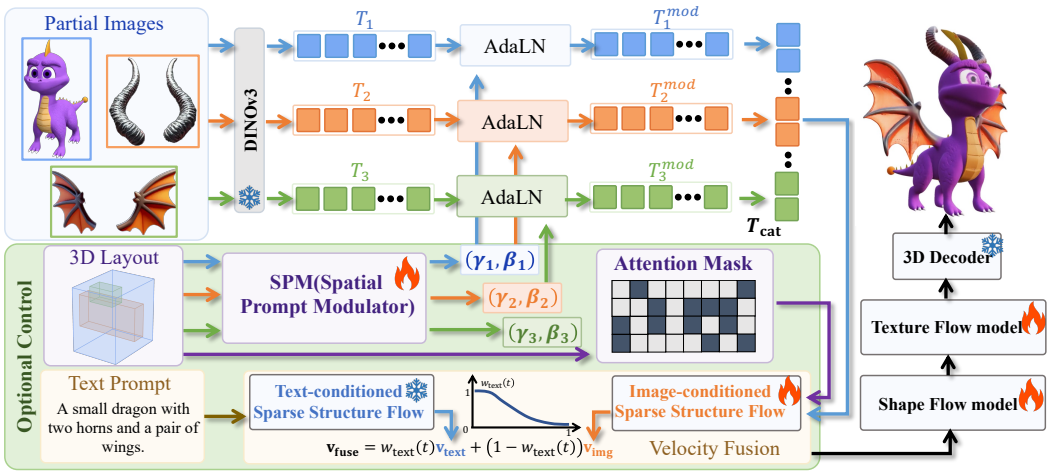

Figure 3: **Overview.** Given multiple partial images, Partial3D generates a coherent 3D asset while supporting optional 3D layout and text controls. We use DINO to encode partial images and concatenate their features into a unified visual-token sequence. The 3D layout is incorporated through our Spatial Prompt Modulator and spatial cross-attention masking, while text descriptions are integrated in a training-free manner via velocity fusion. The sparse-structure result sampled with the fused velocity is then processed by the subsequent 3D backbone to generate the final 3D asset.

to the indirect 2D-to-3D transfer. Later works explore reconstruction-based pipelines Zheng et al. (2025); Erkoç et al. (2025); Li et al. (2025); Gat et al. (2026), which can achieve improved editing quality, but they still largely rely on 2D editing methods and often require manually specified 3D masks to localize the editing region. More recent methods, such as Easy3E Hu et al. (2026), leverage existing flow-based 2D editing techniques Kulikov et al. (2025) and lift the edited results into 3D space, but the editing process is still performed in 2D before being transferred to 3D. Overall, most existing 3D editing methods remain driven by 2D editing models and subsequent 3D lifting, making their control signals predominantly text-based and less visually grounded than direct image-conditioned generation. Concurrent work ES3D Jin et al. (2026b) instead constructs a semantic embedding for an existing 3D asset, retrieves editable components using image or text queries, and modifies them through generative inpainting while preserving the remaining regions. Unlike ES3D, which focuses on semantic component retrieval and local editing of an existing asset, our primary formulation jointly synthesizes a complete asset from independently cropped partial images and subsequently supports editing by manipulating these conditions.

Recent feed-forward native 3D frameworks support multimodal or structured controls Ye et al. (2026); Team (2025); Yan et al. (2026). Most use a holistic image condition, while Fuse3D Jin et al. (2025) extends this setting with multi-image fusion, image-to-3D correspondence, and regional control. Fuse3D initializes the global structure from a holistic image and uses additional image regions to control corresponding areas. In contrast, we jointly infer a coherent global structure from independently cropped partial images, while optionally incorporating layout and text control. Thus, our distinction from Fuse3D lies in the partial-image-only formulation and joint global-structure inference, rather than in multi-image fusion or image-to-3D correspondence itself.

3 Method

We propose Partial3D, which takes partial images with optional 3D bounding boxes and text prompts, and generates controllable 3D assets. In Section 3.1, we review the TRELLIS2 backbone and conditional flow matching formulation. In Section 3.2, we introduce our multi-image spatial grounding design, and in Section 3.3, we present the training-free velocity fusion strategy for multimodal control. Figure 3 shows the overall pipeline of our method.

3.1 Preliminary

Our framework builds upon the TRELLIS2 architecture Xiang et al. (2025a), leveraging its robust generative priors for native 3D synthesis. Within the TRELLIS2 paradigm, a 3D asset is formulated

using the SLAT representation, which parameterizes a 3D scene as a discrete set of sparse, feature-enriched voxels:

$$\boldsymbol{Z} = \{(\boldsymbol{z}_i^{\text{shape}}, \boldsymbol{z}_i^{\text{texture}}, \boldsymbol{p}_i)\}_{i=1}^{L}, \quad \boldsymbol{z}_i \in \mathbb{R}^C,\ \boldsymbol{p}_i \in \{0, 1, \ldots, N-1\}^3, \tag{1}$$

where L represents the total number of active voxels in the scene. For each i-th active voxel, $\boldsymbol{p}_i$ denotes its spatial coordinate index within a dense 3D grid of resolution N. The variables $\boldsymbol{z}_i^{\text{shape}}$ and $\boldsymbol{z}_i^{\text{texture}}$ are C-dimensional latent vectors that encapsulate the local geometric structure and textural details of that specific voxel, respectively. Once generated, this SLAT representation is decoded by a pre-trained 3D Variational Autoencoder (VAE) into an explicit o-voxel format, which can subsequently be transformed into a high-quality mesh.

To synthesize the SLAT representation, TRELLIS2 utilizes three distinct flow-based models to sequentially generate the spatial distributions $\boldsymbol{p}_i$, the shape latents $\boldsymbol{z}_i^{\text{shape}}$, and finally the texture latents $\boldsymbol{z}_i^{\text{texture}}$. These flow-based models are trained via a Conditional Flow-Matching (CFM) objective, where a time-dependent vector field v_θ is learned to regress the target trajectory by optimizing the following loss formulation:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, x_0, \epsilon} \left\| v_\theta(x, t, c) - (\epsilon - x_0) \right\|_2^2, \tag{2}$$

where $t \in [0, 1]$ is the continuous time step, x_0 denotes the clean target data, ϵ represents the noise sampled from $\mathcal{N}(0, 1)$, c denotes the conditioning signal, and $v_\theta(x, t, c)$ is the predicted velocity by the network parameters θ.

3.2 MULTI-IMAGE SPATIAL GROUNDING FOR 3D GENERATION

Given multiple partial images, our goal is to compose them into a coherent 3D asset. We first extract image features and concatenate them into a unified visual-token sequence. Since the spatial relationships among partial images are ambiguous, we introduce SPM and spatial cross-attention masking to optionally ground each image condition in 3D space. Finally, we fine-tune the pretrained image-to-3D backbone in a parameter-efficient manner, using a regularization loss to preserve its original generation prior.

Visual Feature Extraction. To process multiple partial images, we follow the TRELLIS2 Xiang et al. (2025a) architecture and utilize a pre-trained DINOv3 encoder Siméoni et al. (2025) for image encoding. Following Fuse3D Jin et al. (2025), we concatenate DINO features from multiple partial images along the patch dimension into a unified visual-token sequence, which preserves instance-level separation while remaining compatible with cross-attention-based conditioning.

Spatial Prompt Modulator (SPM) To anchor visual tokens to user-specified spatial regions, we introduce the Spatial Prompt Modulator (SPM). For each partial image, we encode its associated 3D bounding boxes following the prompt-encoding strategy of SAM Kirillov et al. (2023), producing a set of spatial prompt embeddings. These embeddings are then aggregated by a learnable query-based pooling module to obtain a compact layout representation for the corresponding image. Crucially, this design naturally supports many-to-one mappings, allowing a single partial image to be associated with multiple 3D bounding boxes simultaneously. The resulting layout representation is passed through an MLP to predict per-image scale γ and shift β parameters. These parameters modulate the corresponding visual tokens following an AdaLN-style formulation:

$$\mathbf{T}_i^{\text{mod}} = \text{LayerNorm}(\mathbf{T}_i) \odot (1 + \gamma_i) + \beta_i, \tag{3}$$

where $\mathbf{T}_i$ denotes the visual tokens of the i-th partial image, and γ_i and β_i are the corresponding modulation parameters. When no bounding box is provided, we use a learned null prompt embedding, allowing SPM to fall back to pure image-conditioned generation without disrupting the pretrained visual feature space.

Spatial Cross-Attention Masking. While the SPM effectively shifts the token distributions to reflect spatial priors, global modulation alone cannot strictly guarantee that visual details from one object part will not leak into unselected regions. To restrict direct cross-attention to the assigned regions, we construct a dense 3D cross-attention mask whenever a 3D bounding box is provided. This masking is applied only during the sparse-structure stage of TRELLIS2, where the overall object structure is determined. During this phase's cross-attention computation, attention logits

corresponding to target voxels outside the assigned 3D bounding box are set to $-\infty$. Conversely, if no bounding box is provided for an image, no mask is applied, allowing those unconstrained visual tokens to naturally attend globally across the entire 3D volume.

Parameter-Efficient Fine-Tuning and Regularization Loss. Rather than training a multimodal 3D generator from scratch, we leverage the powerful generative priors of the pre-trained TRELLIS2 model through targeted fine-tuning. To adapt this flow-based backbone to unstructured partial inputs without degrading its inherent 3D capabilities, we adopt a parameter-efficient fine-tuning strategy inspired by Custom Diffusion Kumari et al. (2023). Our parameter-change analysis motivates fine-tuning only cross-attention and SPM, leaving the remaining backbone frozen (Supplementary Sec. A.2). Furthermore, to mitigate overfitting and catastrophic forgetting during fine-tuning, we employ a joint training objective based on the CFM formulation in Equation 2. At each optimization step, we compute a partial-image flow loss L_{partial} using the extracted partial-image features as the conditioning signal c. In parallel, we compute a regularization loss L_{full} with the full image as the condition, mirroring the original TRELLIS2 training objective. The total objective is formulated as:

$$L_{\text{total}} = L_{\text{partial}} + \lambda L_{\text{full}}, \tag{4}$$

where λ is the regularization weight on L_{full}; we set $\lambda = 1$ in all experiments.

3.3 Velocity Fusion for Multimodal Control

While partial images and 3D layouts provide local control, annotating 3D bounding boxes can be cumbersome; we therefore support text as an optional cue for global structure. We use the text-conditioned sparse-structure flow model from the original TRELLIS architecture, which shares the VAE latent space of our image-conditioned backbone, allowing both velocity predictions to be evaluated in the same latent coordinates. Inspired by compositional diffusion Liu et al. (2022), we exploit the fact that flow-matching models predict the velocity of the latent state x at time t, enabling training-free multimodal synthesis through direct fusion of the velocity predictions from the image-conditioned and text-conditioned models. However, merging these fields with a constant weight throughout the sampling trajectory is sub-optimal, since the text-conditioned model provides coarse semantic guidance and may interfere with the more specific and fine-grained visual details carried by the image conditions. Motivated by observations of coarse-to-fine generation in diffusion models Qian et al. (2024), we assign stronger text guidance early in sampling and reduce it later to favor image-specific details. This is a scheduling heuristic, rather than a verified frequency decomposition of the two velocity fields. We combine the vector fields using a cosine decay of the text velocity weight:

$$\mathbf{v}_{\text{fused}} = (1 - \omega_{\text{text}}(t))\mathbf{v}_{\text{img}} + \omega_{\text{text}}(t)\mathbf{v}_{\text{txt}}, \tag{5}$$

where $\omega_{\text{text}}(t) = (1 - \alpha(t))\omega_{\text{start}} + \alpha(t)\omega_{\text{end}}$ and $\alpha(t) = \frac{1-\cos(\pi t)}{2}$. In all experiments, we set $\omega_{\text{start}} = 1$ and $\omega_{\text{end}} = 0$. Here, $t \in [0, 1]$ denotes normalized sampling progress from start to finish. With $\omega_{\text{start}} > \omega_{\text{end}}$, the schedule progressively reduces the text contribution.

3.4 Native 3D Editing

Our framework naturally supports flexible native 3D editing. Similar to editing text-conditioned image generation models by manipulating text prompts, Partial3D enables users to edit the generated 3D asset by directly manipulating partial image conditions. For example, replacing one partial image with another can modify the corresponding object component, while adding or removing a partial image condition can lead to component-level addition or removal in the generated 3D asset. However, 3D editing requires not only modifying the target region but also preserving the geometry and appearance of the unedited regions Erkoç et al. (2025); Hyung et al. (2023). Inspired by Fuse3D Jin et al. (2025), we leverage the 2D-to-3D correspondence between partial image tokens and generated 3D regions to identify the component associated with each input image. Given an edited set of partial image conditions, we first identify the 3D regions corresponding to the unchanged partial images using the established image-to-3D correspondence, and treat these regions as preservation regions. At each sampling step, we inject source-asset latents at the current timestep's noise level into the preservation regions and apply the predicted flow velocity only to editable regions, as in mask-based generative inpainting Lugmayr et al. (2022). This allows Partial3D to perform native 3D editing while preserving the structure and appearance of regions unrelated to the edit.

Table 1: Quantitative comparison across different methods.

Metric	TRELLIS2	Fuse3D	MUSE	XVerse	Ours
CLIP ↑	0.195	0.206	0.233	0.225	**0.248**
Masked CLIP ↑	0.501	0.563	0.548	0.552	**0.574**
Masked LPIPS ↓	0.692	0.655	0.652	0.654	**0.636**
Masked DINO ↑	0.381	0.392	0.430	0.428	**0.444**
BBox IoU ↑	0.184	0.213	0.418	0.415	**0.682**

Table 2: User study results across different methods. Each criterion is rated on a 1–5 scale.

Criterion	TRELLIS2	Fuse3D	MUSE	XVerse	Ours
Visual Quality ↑	3.42	3.56	3.96	3.74	**4.31**
Partial-image Consistency ↑	2.31	3.12	3.61	3.28	**4.18**
Compositional Coherence ↑	2.48	3.05	3.72	3.26	**4.12**
Control Alignment ↑	2.05	2.74	3.78	3.42	**4.35**

4 EXPERIMENTS

Implementation Details. Each flow model is trained for 12,000 steps with a batch size of 16 on two NVIDIA RTX PRO 6000 GPUs, taking approximately one day. We use the AdamW optimizer with a learning rate of 1×10^{-4}. During training, we randomly drop the 3D bounding-box condition with a probability of 50% to preserve the flexibility of layout-free generation. As described in Section 3.2, we fine-tune only the cross-attention parameters of pretrained TRELLIS2 Xiang et al. (2025a) and the SPM parameters, keeping all other parameters frozen. During inference, we use the TRELLIS2 sampling configuration.

Baselines. Since ***partial image-conditioned 3D generation has not been extensively explored in prior work***, we compare Partial3D with several representative baselines from both controllable 2D generation and native 3D generation. For controllable 2D generation, we include XVerse Chen et al. (2025) and MUSE Peng et al. (2025), which can generate images from multiple visual concepts or spatial layouts. Since these methods operate in the 2D image domain, we further lift their generated images to 3D using TRELLIS2 Xiang et al. (2025a) to construct indirect 2D-to-3D baselines. For native 3D generation, we compare with TRELLIS2 Xiang et al. (2025a), which serves as the backbone image-to-3D model, and Fuse3D Jin et al. (2025), which combines a holistic reference with regional image controls. We provide baseline details and comparisons with a SOTA commercial image generation model OpenAI (2026) in Supplementary Secs. B.4 and C.6.

4.1 COMPARISON RESULTS

4.1.1 QUALITATIVE EVALUATION

Figure 4 shows qualitative comparisons across different methods. Given multiple partial images as input, our method generates 3D content that consistently preserves the visual characteristics of all input parts. In contrast, TRELLIS2, as a standard image-to-3D model, tends to follow only one dominant visual condition and cannot naturally compose multiple partial image inputs. Fuse3D can sometimes incorporate cues from multiple images, as shown in the vase-and-flower example, but it often fails to maintain reasonable part-level geometry or spatial composition. For 2D controllable generation baselines such as MUSE and XVerse, visual information is first injected into a text-to-image backbone and then lifted to 3D. Although this strategy can produce plausible 2D images, the generated 3D assets often exhibit weaker visual consistency with the input partial images. Moreover, the 2D-to-3D lifting pipeline struggles when target components are occluded or not visible from the synthesized front view, such as the dragon tail case, leading to missing or inaccurate 3D structures. These comparisons demonstrate the advantage of directly composing partial image conditions within a native 3D generation framework.

4.1.2 QUANTITATIVE EVALUATION

Table 1 evaluates global semantic alignment with CLIP, part-level visual consistency with Masked CLIP, Masked DINO, and Masked LPIPS, and spatial consistency with 3D bounding-box IoU. Our method achieves the best scores across these metrics. Moreover, we conduct a user study across methods in terms of visual quality, partial-image consistency, compositional coherence, and control



Figure 4: Comparison across different methods.



(a) Spatial prompt modulator **(b)** Cross-attention masking

Figure 5: Ablation study of spatial grounding. (a) The spatial prompt modulator improves alignment with the specified layout. (b) Cross-attention masking constrains local geometry within the target bounding box.

alignment. As shown in Table 2, our method achieves the best performance across all evaluation criteria. Evaluation data and protocols are detailed in Supplementary Sec. B. Additional quantitative comparisons, robustness analyses, and diagnostic experiments are provided in Appendix C.

4.2 Ablation Study

We conduct ablation studies to validate the effectiveness of each design choice in our method, with additional ablations and implementation details provided in Supplementary Sec. C.10.

4.2.1 Training Strategy

Table 3 compares different training strategies by varying the fine-tuned parameter groups and the use of the regularization loss. The results show that cross-attention-only fine-tuning with the regularization loss achieves the best performance across all metrics. Figure 6 further provides visual comparisons under different training strategies. In the first row, we use a single image input to examine whether each strategy preserves the original image-to-3D capability. Removing the regularization loss causes the most severe degradation, while our strategy produces results closest to the original TRELLIS2 model. A similar trend appears in the multi-partial-image setting, where full-parameter fine-tuning tends to overfit and weakens the pretrained image-to-3D prior, leading to less coherent results. In contrast, our cross-attention-only fine-tuning with regularization adapts the model to partial-image conditioning while preserving the original generation capability.

4.2.2 Spatial Prompt Modulator

Table 3 shows that the Spatial Prompt Modulator substantially improves BBox IoU while also improving the other metrics, indicating that it injects 3D layout information and enhances spatial controllability. Figure 5(a) further shows that layout control is better achieved with the proposed SPM.

Table 3: Quantitative results of the ablation study.

Metric	Cross-Attn Tune w/o Reg.	All-Param Tune	Self-Attn Tune	Self+Cross-Attn Tune	w/o SPM	w/o Cross-Attn Masking	Cross-Attn Tune w/ Reg. (Ours)
CLIP ↑	0.229	0.234	0.226	0.234	0.224	0.240	**0.248**
Masked CLIP ↑	0.565	0.564	0.551	0.562	0.558	0.559	**0.574**
Masked LPIPS ↓	0.655	0.656	0.651	0.653	0.648	0.655	**0.636**
Masked DINO ↑	0.406	0.377	0.395	0.400	0.350	0.400	**0.444**
BBox IoU ↑	0.575	0.577	0.571	0.570	0.493	0.557	**0.682**

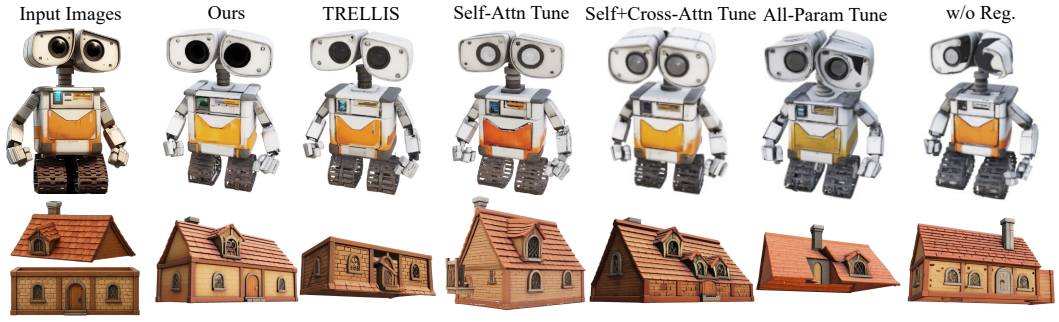

Figure 6: Ablation study of the training strategy.

4.2.3 Cross-Attention Masking Strategy

Table 3 also validates the effectiveness of the cross-attention masking strategy. With the mask, the generated content follows the specified 3D bounding-box layout more accurately, better matching the intended layout. As shown in Figure 5(b), removing the mask can cause local geometry, such as the Santa hat, to grow outside the target bounding box, while our method keeps the structure better constrained within the assigned region.

4.3 Applications

Shell-like Geometry With 3D bounding-box conditions, our model can effectively alleviate the shell-like geometry problem that often appears in image-to-3D generation from flat-colored 2D images, which has also been observed in prior works Cong et al. (2026); Team (2025). As shown in Figure 13, injecting 3D bounding-box information through our SPM provides additional spatial priors and helps generate more plausible volumetric 3D assets.

Flexible Control and Native 3D Editing By treating partial images as separate conditioning signals, our framework enables native 3D editing. In addition, optional layout and text conditions further enhance the controllability of 3D generation. As shown in Figure 14, Partial3D can edit generated 3D assets by simply replacing one partial image condition with another. It can also control the spatial arrangement of the target 3D asset through 3D layout injection, or guide the global structure using a text description. Furthermore, component-level addition and removal can be naturally achieved by adding or removing the corresponding partial image conditions.

5 Conclusion

In this paper, we propose Partial3D, a native 3D generative model for controllable synthesis from partial observations. Unlike existing 3D control methods that often rely on indirect 2D-to-3D pipelines, Partial3D directly composes 3D assets from multiple partial images with optional 3D layout and text controls. To preserve the pretrained image-to-3D prior, we adopt parameter-efficient fine-tuning with a regularization loss, and further introduce SPM for spatial grounding and training-free velocity fusion for text guidance. By treating partial images as separable compositional units, analogous to words in a text prompt, Partial3D enables native 3D flow-based editing through image replacement and provides a unified framework for partial-image-conditioned 3D generation with optional text and layout control.

AI USE STATEMENT

We used AI tools to assist with LaTeX document formatting and grammar correction. We also used AI tools to support dataset construction, including caption generation. All AI-assisted outputs were reviewed by the authors.

REPRODUCIBILITY STATEMENT

We describe the model architecture and proposed components in Sections 3.2 and 3.3. The experimental section reports the training configuration, baselines, and evaluation protocol. The appendix provides additional implementation details, training-data preparation, runtime analysis, baseline configurations, metric definitions, and ablations. Together, these details are intended to support independent reproduction of our method and evaluation. We will publicly release all source code and datasets associated with this work.

REFERENCES

Black Forest Labs. FLUX.1-dev. https://huggingface.co/black-forest-labs/FLUX.1-dev, 2024. Hugging Face model page. Accessed: 2026-05-11.

Blender Online Community. Blender – a 3d modelling and rendering package, 2018. URL https://www.blender.org.

Bowen Chen, Mengyi Zhao, Haomiao Sun, Li Chen, Xu Wang, Kang Du, and Xinglong Wu. Xverse: Consistent multi-subject control of identity and semantic attributes via dit modulation. *arXiv preprint arXiv:2506.21416*, 2025.

Yankuan Chi, Xiang Li, Zixuan Huang, and James M Rehg. Vinedresser3d: Agentic text-guided 3d editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, June 2026.

Xiaoyan Cong, Jiayi Shen, Zekun Li, Rao Fu, Tao Lu, and Srinath Sridhar. Art3d: Training-free 3d generation from flat-colored illustration, 2026. URL https://arxiv.org/abs/2504.10466.

Yusuf Dalva, Guocheng Gordon Qian, Maya Goldenberg, Tsai-Shien Chen, Kfir Aberman, Sergey Tulyakov, Pinar Yanardag, and Kuan-Chieh Jackson Wang. Canvas-to-image: Compositional image generation with multimodal controls, 2025. URL https://arxiv.org/abs/2511.21691.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A universe of 10m+ 3d objects. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=Sq3CLKJeiz.

Ziya Erkoç, Can Gümeli, Chaoyang Wang, Matthias Nießner, Angela Dai, Peter Wonka, Hsin-Ying Lee, and Peiye Zhuang. Preditor3d: Fast and precise 3d shape editing. In *CVPR*, pp. 640–649, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Erkoc_PrEditor3D_Fast_and_Precise_3D_Shape_Editing_CVPR_2025_paper.html.

Daniel Garibi, Shahar Yadin, Roni Paiss, Omer Tov, Shiran Zada, Ariel Ephrat, Tomer Michaeli, Inbar Mosseri, and Tali Dekel. Tokenverse: Versatile multi-concept personalization in token modulation space. *ACM Trans. Graph.*, 44(4), July 2025. ISSN 0730-0301. doi: 10.1145/3730843. URL https://doi.org/10.1145/3730843.

Inbar Gat, Dana Cohen-Bar, Guy Levy, Elad Richardson, and Daniel Cohen-Or. Shapeup: Scalable image-conditioned 3d editing, 2026. URL https://arxiv.org/abs/2602.05676.

Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Shimin Hu, Yuanyi Wei, Fei Zha, Yudong Guo, and Juyong Zhang. Easy3e: Feed-forward 3d asset editing via rectified voxel flow. *arXiv preprint arXiv:2602.21499*, 2026.

Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly DDPM noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12469–12478, 2024.

Junha Hyung, Sungwon Hwang, Daejin Kim, Hyunji Lee, and Jaegul Choo. Local 3d editing via 3d distillation of clip knowledge. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12674–12684, 2023.

Li Jin, Weikai Chen, Yujie Wang, Yingda Yin, Zeyu Hu, Runze Zhang, Keyang Luo, Shengju Qian, Xin Wang, and Xueying Qin. Cosmo3d: Open-world promptable 3d semantic part segmentation through llm-guided canonical spatial modeling. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, June 2026a.

Xuancheng Jin, Rengan Xie, Wenting Zheng, Rui Wang, Hujun Bao, and Yuchi Huo. Fuse3d: Generating 3d assets controlled by multi-image fusion. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, SA Conference Papers '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400721373. doi: 10.1145/3757377.3763943. URL https://doi.org/10.1145/3757377.3763943.

Xuancheng Jin, Rengan Xie, Jiayuan Lu, Wenting Zheng, Rui Wang, Lincheng Li, Yingfeng Chen, and Yuchi Huo. ES3D: Component-aware 3d editing via 3d semantic embedding, 2026b. URL https://arxiv.org/abs/2608.15749.

Jeongsol Kim, Yeobin Hong, Jonghyun Park, and Jong Chul Ye. Flowalign: Trajectory-regularized, inversion-free flow-based image editing. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nyttIJfwW7.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19721–19730, 2025.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Qingxiang Lin, Jingwei Huang, Chunchao Guo, and Xiangyu Yue. Lattice: Democratize high-fidelity 3d generation at scale, 2025. URL https://arxiv.org/abs/2512.03052.

Peng Li, Suizhi Ma, Jialiang Chen, Yuan Liu, Chongyi Zhang, Wei Xue, Wenhan Luo, Alla Sheffer, Wenping Wang, and Yike Guo. Cmd: Controllable multiview diffusion for 3d editing and progressive generation. *arXiv preprint arXiv:2505.07003*, 2025.

Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: Text-driven 3d editing via focal-fusion assembly, 2023. URL https://arxiv.org/abs/2308.10608.

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. In *International Conference on Computer Vision (ICCV)*, 2025.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pp. 423–439. Springer, 2022.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models, 2022. URL https://arxiv.org/abs/2211.09794.

OpenAI. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

OpenAI. ChatGPT Images 2.0 is here, April 2026. URL https://openai.com/zh-Hans-CN/index/introducing-chatgpt-images-2-0/.

Gaurav Parmar, Or Patashnik, Kuan-Chieh Wang, Daniil Ostashev, Srinivasa Narasimhan, Jun-Yan Zhu, Daniel Cohen-Or, and Kfir Aberman. Object-level visual prompts for compositional image generation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

Fei Peng, Junqiang Wu, Yan Li, Tingting Gao, Di Zhang, and Huiyuan Fu. Muse: Multi-subject unified synthesis via explicit layout semantic expansion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15885–15895, 2025.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. URL https://arxiv.org/abs/2307.01952.

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

Yurui Qian, Qi Cai, Yingwei Pan, Yehao Li, Ting Yao, Qibin Sun, and Tao Mei. Boosting diffusion models with moving average sampling in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8911–8920, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Hu0FSOSEyS.

Etai Sella, Hao Phung, Nitay Amiel, Or Litany, Or Patashnik, and Hadar Averbuch-Elor. Prox-e: Fine-grained 3d shape editing via primitive-based abstractions. *arXiv preprint arXiv:2604.23774*, 2026.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Yukai Sun, Hao Qin, Ming Kong, Luyuan Chen, Zhijie Xu, Jinjian Zhang, Jie Liu, Feng Zhang, and Qiang Zhu. CAST3D: Customizing arbitrary 2d assets into 3d world. In *Computer Vision – ECCV 2026*, pp. 446–464. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-37038-9_25. URL https://doi.org/10.1007/978-3-032-37038-9_25.

Gemma C Tarrés, Zhe Lin, Zhifei Zhang, He Zhang, Andrew Gilbert, John Collomosse, and Soo Ye Kim. Multitwine: Multi-object compositing with text and layout control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Tencent Hunyuan3D Team. Hunyuan3d-omni: A unified framework for controllable generation of 3d assets, 2025. URL https://arxiv.org/abs/2509.21245.

Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation, 2024.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=ppJuFSOAnM.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, et al. Native and compact structured latents for 3d generation. *arXiv preprint arXiv:2512.14692*, 2025a.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 21469–21480, June 2025b.

Yuhuan Xie, Aoxuan Pan, Mingxian Lin, Wei Huang, Yi-Hua Huang, and Xiaojuan Qi. 2d instance editing in 3d space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1967–1974, 2025.

Hongyu Yan, Kunming Luo, Weiyu Li, Kaiyi Zhang, Yixun Liang, Jingwei Huang, Chunchao Guo, and Ping Tan. Posemaster: A unified 3d native framework for stylized pose generation, 2026. URL https://arxiv.org/abs/2506.21076.

Chongjie Ye, Cheng Cao, Chuanyu Pan, Yiming Hao, Yihao Zhi, Yuanming Hu, and Xiaoguang Han. Omni123: Exploring 3d native foundation models with limited 3d data by unifying text to 2d and 3d generation. *arXiv preprint arXiv:2604.02289*, 2026.

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), July 2023. ISSN 0730-0301. doi: 10.1145/3592442. URL https://doi.org/10.1145/3592442.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Yang Zheng, Mengqi Huang, Nan Chen, and Zhendong Mao. Pro3d-editor: A progressive-views perspective for consistent and precise 3d editing. *arXiv preprint arXiv:2506.00512*, 2025.

Dewei Zhou, You Li, Fan Ma, Zongxin Yang, and Yi Yang. Migc++: Advanced multi-instance generation controller for image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1714–1728, 2024.

A Additional Method Details

A.1 Spatial Prompt Modulator

In this section, we provide additional implementation details of the Spatial Prompt Modulator (SPM). Given the visual tokens of each partial image, SPM predicts per-image AdaLN modulation parameters from its associated 3D bounding-box prompts. This allows the model to inject spatial layout information into the image conditions while keeping the visual tokens of different partial images separately controllable.

Bounding-box prompt encoding. For each partial image, we allow zero, one, or multiple associated 3D bounding boxes. Each 3D bounding box is represented by its two corner coordinates in the normalized object coordinate system. Following the prompt encoding strategy of SAM Kirillov et al. (2023), we encode each box corner coordinate using random Fourier positional embeddings and add a learnable corner-type embedding to distinguish the two corners. When a partial image is associated with multiple bounding boxes, we concatenate all box-corner embeddings into a single prompt-token sequence. Since the subsequent cross-attention pooling operates over the token dimension, this design naturally supports a variable number of bounding boxes for each partial image.

Query-based prompt pooling. We aggregate the prompt-token sequence into a compact layout representation using a learnable query-based pooling module. Specifically, the prompt tokens are projected into keys and values, and a learnable query attends to them through cross-attention:

$$\mathbf{s}_i = \text{Attention}\left(\mathbf{q}, \mathbf{K}_i, \mathbf{V}_i\right), \tag{6}$$

where $\mathbf{q}$ is the learnable query, and $\mathbf{K}_i$ and $\mathbf{V}_i$ are the keys and values projected from the prompt tokens of the i-th partial image. The resulting vector $\mathbf{s}_i$ serves as the layout representation for that image.

Null prompt for layout-free generation. The 3D bounding-box condition is optional. When no bounding box is provided for a partial image, we use a learned null prompt embedding as its layout representation. This null embedding is processed by the same normalization and modulation layers as regular spatial prompts. As a result, SPM produces default modulation parameters for layout-free inputs, allowing the model to fall back to pure image-conditioned generation while keeping the feature distribution close to that of the pretrained image-to-3D backbone.

AdaLN modulation. The layout representation is normalized and passed through a lightweight MLP to predict per-image modulation parameters:

$$[\gamma_i, \beta_i] = \text{MLP}(\text{LN}(\mathbf{s}_i)), \tag{7}$$

where $\gamma_i, \beta_i \in \mathbb{R}^C$ and C is the image-token dimension. The predicted parameters are then used to modulate the visual tokens of the corresponding partial image:

$$\mathbf{T}_i^{\text{mod}} = \text{LayerNorm}(\mathbf{T}_i) \odot (1 + \gamma_i) + \beta_i. \tag{8}$$

We initialize the last layer of the modulation MLP to zero, so that $\gamma_i = \beta_i = 0$ and the initial output is $\text{LayerNorm}(\mathbf{T}_i)$. This introduces no layout-dependent affine modulation at initialization.

A.2 Parameter-Efficient Fine-Tuning

As described in the main paper, we fine-tune only a small subset of the pretrained flow model to adapt it to partial-image conditioning. To identify which modules are most relevant for this adaptation, we first train the model with all parameters unfrozen and compare the fine-tuned weights with the pretrained checkpoint. For each module group, we report the number of parameters and the relative parameter change in Table 4.

The cross-attention layers exhibit the largest relative change, despite accounting for only 18.28% of the parameters. This indicates that cross-attention is the primary module used by the pretrained backbone to adapt to our new conditioning format, which is consistent with the finding of Custom Diffusion Kumari et al. (2023). Therefore, in our final model, we fine-tune only the cross-attention layers and the newly introduced SPM, while keeping the rest of the pretrained backbone frozen.



Figure 7: More generation results.

Table 4: Parameter-change analysis after full-parameter fine-tuning.

Module Group	Params	Param. Ratio	Rel. Change
Self-Attention	283.39M	21.93%	0.192
Cross-Attention	236.21M	18.28%	**0.215**
Others	772.58M	59.79%	0.198

B Experimental Details

B.1 Training Data Preparation

We construct our training data based on the Objaverse-XL Deitke et al. (2023) assets curated by TRELLIS-500K Xiang et al. (2025b). For each 3D asset, we render 16 RGB views in Blender Blender Online Community (2018) under randomly sampled camera field-of-views and lighting conditions. These rendered images serve as the full-image conditions used for the regularization loss described in the main paper. To obtain partial image conditions, we segment each 3D asset into semantic or geometric parts using PartField Liu et al. (2025). For each segmented 3D part, we project its visible region onto the rendered views to obtain the corresponding 2D masks. Since each 2D mask is projected from a known 3D part, it is naturally paired with the 3D bounding box computed from the same part mask in the normalized object coordinate system. During training, we first randomly sample a rendered view and then randomly select one 2D part mask from that view. We use the selected mask and its inverse to construct two complementary partial images, providing the model with partial observations of the same 3D asset. Before image encoding, we normalize each partial image independently by cropping its tight foreground bounding box, adding padding equal to 8% of the longer foreground side on all sides, and padding the result to a square with a white background. This preprocessing removes the original canvas coordinates and relative scales of the two complementary regions. This strategy encourages the model to compose information from separate visual conditions. Although each training sample contains two complementary conditions from the same object and view, our token-concatenation design accepts a variable number of partial images at inference. The results with three and four input parts in Figure 18 and Table 11 demonstrate generalization beyond the two-part training configuration.

B.2 Evaluation Dataset

Our evaluation set contains 100 examples: 30 examples held out from the original data pool before fine-tuning and 70 examples synthesized using GPT-4o. The 30 held-out examples follow the same-object complementary-pair construction used for training but are never used for model optimization. In contrast, within each synthetic example, the partial input images are independently generated from different source objects rather than obtained by splitting a single complete synthetic object; GPT-4o also generates the corresponding text description. Independently of this data-source split, the 100 examples are partitioned into three mutually exclusive conditioning subsets: 30 image-only cases, 34 image-and-text cases, and 36 image-text-layout cases. This produces an approximately balanced distribution across the three conditioning settings. Within each conditioning subset, the examples cover diverse object categories and compositional configurations. All methods in the main comparison are evaluated on the same 100 examples.

B.3 Runtime Analysis

Our method introduces negligible additional runtime compared with TRELLIS2. During inference, the only extra module in the image-conditioned pipeline is the lightweight Spatial Prompt Modulator, which predicts per-image modulation parameters from 3D bounding-box prompts. Since this module is small compared with the pretrained flow backbone, its computational overhead is negligible. In practice, generating one 3D asset takes less than 15 seconds on a single NVIDIA RTX PRO 6000 GPU, which is comparable to TRELLIS2 under the same sampling configuration. Increasing the number of partial images has little impact on the total generation time, since the dominant cost still comes from the flow backbone rather than the prompt modulation module. When the optional text velocity fusion strategy is enabled, the runtime increases by about 5 seconds. This is because,



Figure 8: Intermediate 2D generation results of 2D baselines.

at each sampling step of the sparse-structure stage, both the image-conditioned and text-conditioned flow models are run to predict their velocities for fusion. The subsequent shape and texture generation stages remain unchanged.

B.4 Baseline Implementation

We compare our method with representative native 3D generation and controllable 2D generation baselines. For the main comparison in Table 1, every method receives the same partial images, while an auxiliary condition is supplied only when it is present in that case and supported by the method. Thus, Table 1 aggregates all 100 cases while respecting both the condition available in each case and the modalities supported by each architecture. Supplementary Sec. C.3 separately compares all methods under matched inputs, using their native null conditions for unavailable auxiliary modalities. For 2D generation baselines, the generated images are further reconstructed into 3D assets using TRELLIS2, so that all methods are evaluated in the same 3D output space. We additionally compare all five methods on the 30 image-only cases (Supplementary Sec. C.3). CAST3D Sun et al. (2026), the method most closely related to our setting, is not included because its official implementation is not publicly available.

Fuse3D We use the official implementation of Fuse3D Jin et al. (2025). Fuse3D is designed to initialize the global structure from a holistic reference image and fuse additional regional image controls through token concatenation. Because our setting does not provide a holistic reference, we feed Fuse3D the same independently cropped partial images used by all methods, without supplying an additional global image.

MUSE We implement MUSE Peng et al. (2025) using the official code and checkpoint built upon SDXL-base-1.0 Podell et al. (2023), as suggested by the official project page. Since MUSE operates in the 2D image domain, we first use it to synthesize a 2D image from the given visual conditions and text description. The text description is generated by GPT-4o OpenAI (2024) to summarize the desired composition from the input partial images. The resulting 2D image is then reconstructed into a 3D asset using TRELLIS2.

Xverse We implement XVerse Chen et al. (2025) using the official code and the FLUX.1-dev Black Forest Labs (2024) backbone, following the recommendation of the official project page. Similar to MUSE, XVerse generates a 2D image conditioned on the input visual concepts and a text description generated by GPT-4o OpenAI (2024). We then lift the generated 2D image to 3D using TRELLIS2 to obtain the final 3D asset for evaluation.

TRELLIS2 We implement TRELLIS2 Xiang et al. (2025a) using the official code and checkpoints. To adapt TRELLIS2 to the multi-image setting, we follow its original multi-image generation strategy from TRELLIS Xiang et al. (2025b). Specifically, at each sampling step, the image-conditioned flow model predicts one velocity for each input partial image, and these velocities are averaged to obtain the final velocity used for updating the latent state. This provides a direct multi-image TRELLIS2 baseline without additional training.

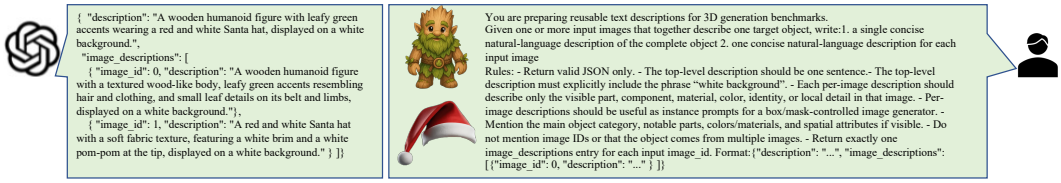

Figure 9: Illustration of our text-description generation process.

B.5 Metrics

CLIP We adopt the ViT-L/14 variant of CLIP Radford et al. (2021) to evaluate the global semantic consistency of the generated 3D assets. For each generated asset, we render multiple views and compute the CLIP similarity between the rendered images and the overall text description of the target asset. The text descriptions are generated by GPT-4o OpenAI (2024) based on the input partial images.

Part-level Image Metrics Masked CLIP, Masked LPIPS, and Masked DINO share the same region-matching protocol. For each input partial image, we identify its corresponding region in the generated asset using Cosmo3D Jin et al. (2026a) and render the region from four viewpoints. ChatGPT-4o selects the rendered view that best matches the input. The input and selected rendering are independently normalized in scale and centered, without pixel-level registration or explicit viewpoint alignment. All Cosmo3D segmentations and GPT-4o-selected views in the evaluation set were manually verified before metric computation to ensure valid part correspondences. Masked CLIP computes cosine similarity between CLIP ViT-L/14 image embeddings, while Masked LPIPS computes the LPIPS-VGG perceptual distance Zhang et al. (2018). For Masked DINO, we use DINOv3 ViT-L/16 (LVD-1689M) Siméoni et al. (2025) and extract its 1,024-dimensional CLS token after the final LayerNorm. We L2-normalize the two CLS features and compute their dot product, equivalent to cosine similarity. The reported scores are averaged over all evaluated partial-image conditions.

3D BBox IoU We use 3D bounding-box IoU to evaluate spatial controllability. For each input partial image with a target 3D bounding box available for evaluation, we identify the corresponding generated 3D region using Cosmo3D Jin et al. (2026a). We then compute the axis-aligned 3D bounding box of the generated region and measure its IoU with the target 3D bounding box.

B.6 User Study Protocol

We recruited 80 participants, each of whom evaluated 6 cases sampled from our 100-example evaluation set. For each case, participants were shown the input partial images, optional control signals, and anonymized outputs from all compared methods in four views under identical camera and rendering settings. The case order and method order were randomized independently. Participants rated visual quality, partial-image consistency, compositional coherence, and control alignment on a 1–5 Likert scale, and Table 2 reports the mean scores across all collected ratings.



Figure 10: Comparison with a commercial 2D image generation model under occlusion. GPT-Image-2 can synthesize visually plausible 2D compositions, which are then lifted to 3D using TRELLIS2. However, the single-view 2D observation fails to reveal occluded structures, leading to incomplete or shell-like geometry after 3D reconstruction.

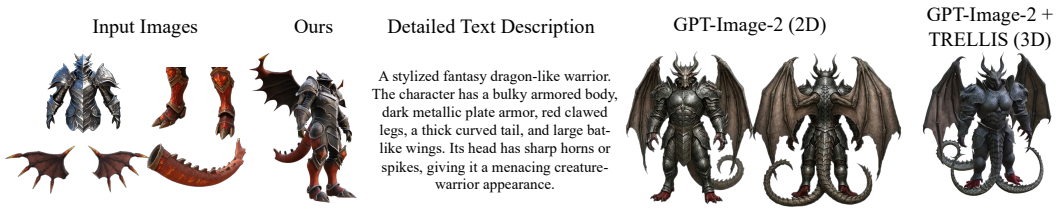

Figure 11: Comparison with a commercial 2D image generation model using detailed text descriptions. Although GPT-Image-2 can generate plausible 2D views from text, the text prompt cannot fully specify the fine-grained visual identity of each input component. Moreover, inconsistencies across independently generated multi-view images can accumulate during TRELLIS reconstruction and degrade the final 3D asset.

B.7 Text Description Generation

We use text descriptions for text-conditioned baselines and CLIP-based evaluation. Figure 9 illustrates the GPT-4o OpenAI (2024) captioning procedure, which produces one concise overall description of the target 3D asset and one concise description for each input partial image. The overall description provides the text condition for text-guided baselines and the reference for global CLIP similarity, while per-image descriptions specify individual visual conditions for methods such as MUSE Peng et al. (2025). In the image-only comparison, evaluation descriptions are not supplied to the generators.

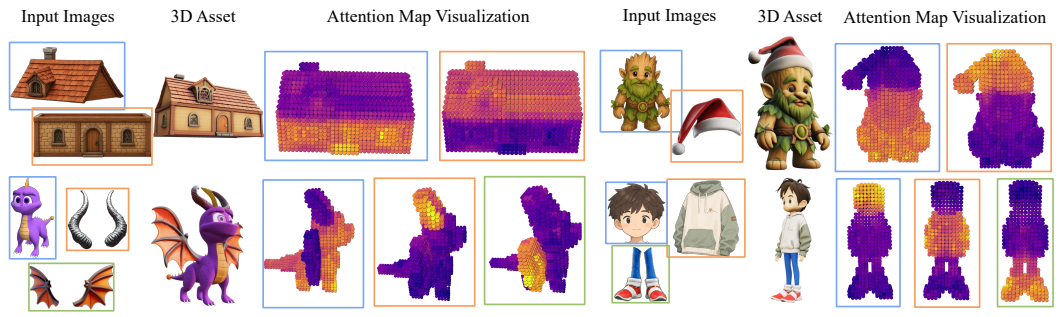

Figure 12: Visualization of the attention map calculated between input partial images and generated 3D regions.

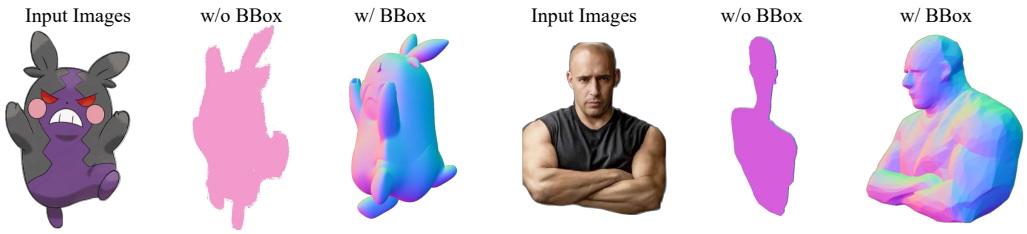

Figure 13: Effect of 3D bounding-box guidance on flat or weakly geometric image inputs.

C Additional Experiments and Discussion of Technical Design

C.1 Applications

Shell-like Geometry With 3D bounding-box conditions, our model can effectively alleviate the shell-like geometry problem that often appears in image-to-3D generation from flat-colored 2D images, which has also been observed in prior works Cong et al. (2026); Team (2025). As shown in Figure 13, injecting 3D bounding-box information through our SPM provides additional spatial priors and helps generate more plausible volumetric 3D assets.

Flexible Control and Native 3D Editing By treating partial images as separate conditioning signals, our framework enables flexible native 3D editing. In addition, optional layout and text conditions further enhance the controllability of 3D generation. As shown in Figure 14, Partial3D can edit generated 3D assets by simply replacing one partial image condition with another. It can also control the spatial arrangement of the target 3D asset through 3D layout injection, or guide the global structure using a text description. Furthermore, component-level addition and removal can be naturally achieved by adding or removing the corresponding partial image conditions.

Partial3D is formulated and trained primarily as a controllable 3D generative model; native editing is an important downstream application enabled by its separable partial-image conditions rather than the primary task formulation. The publicly available 3D editing methods most relevant to our setting rely on a separately edited 2D image to guide the subsequent 3D edit or reconstruction. Their performance therefore also depends on an external 2D editing stage, whereas Partial3D directly replaces a partial-image condition within the native 3D generation process. ES3D Jin et al. (2026b) is more closely related because it supports image-guided component retrieval and native 3D inpainting, but we do not include it in the experimental comparison because its official implementation are not publicly available.

C.2 Editing Effectiveness and Unedited-Region Preservation

We use GPT-4o OpenAI (2024) to construct 30 image-replacement editing cases, each combining partial-image conditions from different source objects. For each case, we evaluate both preservation and adherence to the replacement image, comparing variants with and without source-latent injection on the same source assets, replacement images, and sampling seeds. We obtain the edited and unedited regions using the Fuse3D attention-based image-to-3D correspondence described in Section 3.4 and analyzed in Supplementary Section C.4 and Figure 12. Unedited-region stability is measured by symmetric Chamfer distance in the shared source coordinate system and by LPIPS-VGG and DINOv3 similarity between renderings under identical cameras. To measure editing effectiveness, we use *Target DINO Gain*: the edited region's DINO similarity to the replacement image minus the corresponding pre-edit similarity. All image-based metrics use the rendering, normalization, feature extraction, and aggregation protocol defined in Supplementary Section B.5, with fixed preprocessing and view-selection results for identical regions. An unchanged result has zero gain, while a positive gain indicates improved target adherence; this is a continuous proxy, not a binary success rate. As shown in Table 5, source-latent injection substantially improves unedited-region preservation while maintaining a comparable positive Target DINO Gain in the edited region.

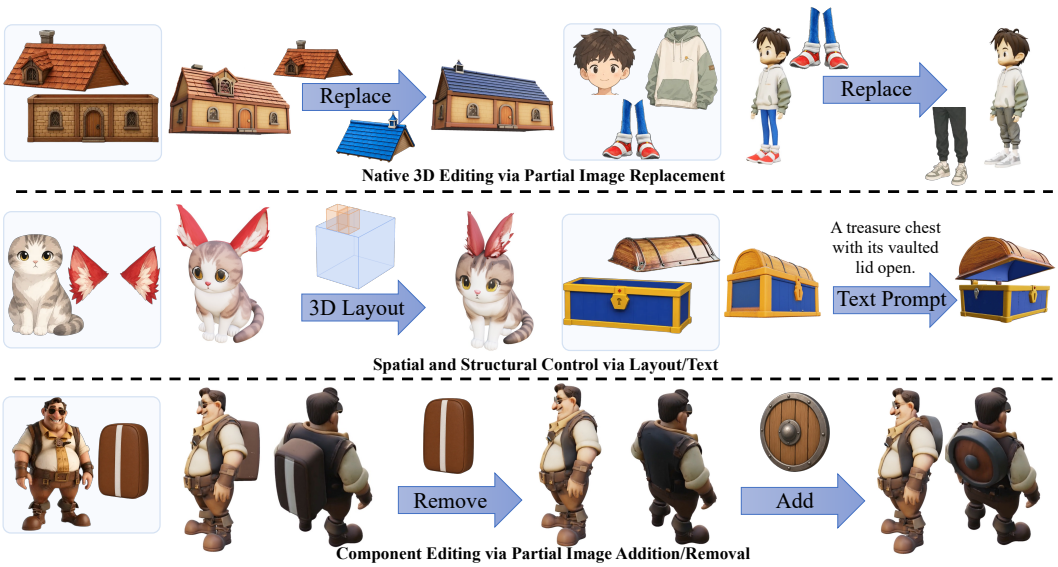

Figure 14: Additional examples of controllable 3D generation and native editing enabled by partial image conditions.

Table 5: Unedited-region preservation and editing effectiveness with and without source-latent injection.

Metric	w/o Injection	w/ Injection
Unedited-region Chamfer Distance ↓	0.028	0.006
Unedited-region LPIPS ↓	0.214	0.073
Unedited-region DINO ↑	0.812	0.948
Edited-region Target DINO Gain ↑	0.118	0.112

C.3 Matched-Condition Comparisons

We evaluate TRELLIS2, Fuse3D, MUSE, XVerse, and Partial3D on the 30 image-only cases. All methods receive the same partial images and no informative text or layout conditions. For MUSE and XVerse, we encode an empty prompt using the models' native unconditional text-conditioning path; MUSE additionally receives no valid layout boxes. We do not use an all-zero text embedding or a zero-coordinate box, since these may constitute out-of-distribution features or an unintended degenerate spatial condition. For Partial3D, text velocity fusion is disabled and the learned null layout prompt is used. Because these image-only cases do not specify target layouts, 3D bounding-box IoU is not reported. Table 6 compares the methods using the metrics defined in Supplementary Sec. B.5, isolating partial-image composition from the effects of additional control signals.

Table 6: Comparison on the 30 image-only cases.

Metric	TRELLIS2	Fuse3D	MUSE	XVerse	Ours
CLIP ↑	0.194	0.205	0.229	0.222	**0.238**
Masked CLIP ↑	0.503	0.561	0.543	0.550	**0.570**
Masked LPIPS ↓	0.691	0.656	0.652	0.655	**0.640**
Masked DINO ↑	0.383	0.394	0.432	0.421	**0.443**

We further perform two pairwise comparisons with matched supported conditions. On the 36 image-text-layout cases, MUSE and Partial3D receive the same partial images and text descriptions; the target 3D boxes are supplied directly to Partial3D and projected into 2D for MUSE. On the 34 text-conditioned cases, XVerse and Partial3D receive the same partial images and text descriptions

without layout inputs. As shown in Table 7, BBox IoU is reported only for the layout-conditioned comparison.

Table 7: Matched-condition comparisons. The left block compares image+text+layout conditioning, with 2D projected boxes for MUSE and native 3D boxes for Ours. The right block compares image+text conditioning.

Image + Text + Layout			Image + Text		
Metric	MUSE	Ours	Metric	XVerse	Ours
CLIP ↑	0.242	**0.255**	CLIP ↑	0.236	**0.251**
Masked CLIP ↑	0.552	**0.579**	Masked CLIP ↑	0.554	**0.573**
Masked LPIPS ↓	0.650	**0.632**	Masked LPIPS ↓	0.653	**0.637**
Masked DINO ↑	0.434	**0.448**	Masked DINO ↑	0.431	**0.441**
BBox IoU ↑	0.418	**0.682**			

C.4 Attention and Correspondence Analysis

To better understand how partial-image conditions are grounded in the generated 3D asset, we visualize the cross-attention responses between image tokens and sparse voxels. As shown in Figure 12, different partial images attend to different regions of the generated 3D structure. For example, the hat image produces strong responses on the hat region, while the character image mainly attends to the body region. This indicates that our model does not simply merge all visual inputs into a holistic condition, but instead learns part-level correspondences between individual partial images and their associated 3D regions. Such behavior supports our design of treating partial images as separable visual conditions for controllable 3D generation and editing.

C.5 Geometric Evaluation on Held-Out Assets

In Table 8, we additionally evaluate all methods on the 30 cases whose source assets were held out at the asset level before fine-tuning and therefore have ground-truth meshes. None of these source assets appears in our fine-tuning set. We use the same input protocol as in the main comparison and adopt the CD and F1 definitions used by TRELLIS2. After normalizing the generated and reference meshes to a unit cube, we uniformly sample 1,000,000 points from each mesh surface, matching the sample count used by TRELLIS2. We report symmetric squared-L_2 Chamfer Distance (CD), scaled by 10^6, and F1 Score with the squared-distance threshold $\tau = 10^{-6}$. This subset-specific evaluation complements the image-based metrics on the full 100-case benchmark.

Table 8: Geometric comparison on the cases with reference meshes.

Metric	TRELLIS2	Fuse3D	MUSE	XVerse	Ours
CD ($\times 10^6$) ↓	239.84	181.67	83.21	133.05	**13.31**
$F1_{10^{-6}}$ ↑	0.212	0.328	0.574	0.189	**0.661**

C.6 Comparison with a Commercial 2D Generation Model

In this section, we further compare our method with a state-of-the-art commercial 2D generation and editing model, GPT-Image-2 OpenAI (2026). We acknowledge that such 2D image models are highly powerful and can be flexibly controlled by multimodal inputs. However, when they are used in a 2D-generation-then-lifting pipeline for 3D asset creation or editing, the generated 2D images still serve only as 2D observations. As also discussed in prior works Ye et al. (2026); Sella et al. (2026); Chi et al. (2026); Xie et al. (2025), such indirect pipelines may suffer from multi-view inconsistency, object-identity drift, and limited preservation of unedited 3D regions.

We first evaluate whether detailed text descriptions can replace explicit partial-image conditions. As shown in Figure 11, even when GPT-Image-2 is given a detailed text description, the generated views may not faithfully preserve all fine-grained visual details from the input images, such as the exact



Figure 15: Robustness to occluded and rotated partial-image observations.

Table 9: Robustness to partial observations.

Metric	Ours	15% Occluded	50% Occluded	75% Occluded	Synthetic 45° viewpoint
CLIP ↑	**0.241**	0.236	0.234	0.236	0.224
BBox IoU ↑	**0.682**	0.680	0.674	0.679	0.642
Masked CLIP ↑	**0.568**	0.542	0.530	0.520	0.539
Masked LPIPS ↓	**0.642**	0.685	0.683	0.684	0.664
Masked DINO ↑	**0.446**	0.423	0.410	0.397	0.384

armor style, red clawed legs, wing shape, and tail appearance. Moreover, inconsistencies across independently generated views can accumulate during 3D reconstruction, degrading the quality of the final asset. In contrast, our method directly uses partial images as visual conditions, allowing the generated 3D asset to better preserve the specific appearance of each input component.

We further study occlusion cases in Figure 10. GPT-Image-2 can produce visually appealing 2D compositions from the input images. However, under object occlusion, the single generated image does not provide sufficient information to recover the complete 3D structure. For example, in the turtle case, the mushroom shell occludes the turtle body, causing the subsequent TRELLIS reconstruction to produce incomplete or shell-like internal geometry in the sectional view. Similarly, in the flowerpot case, the plant is partially hidden inside the pot, and the 2D-to-3D pipeline struggles to infer the correct occluded structure inside the container. These results highlight that, even with strong commercial 2D generators, indirect 2D-to-3D pipelines cannot fully close the gap to native 3D generative models for visually specific, part-level, and occlusion-aware 3D composition.

C.7 Robustness to Occlusion and Viewpoint Variation

To evaluate robustness to imperfect partial observations in real-world scenarios, we consider both occlusion and viewpoint variation. For occlusion, we randomly mask 15%, 50%, and 75% of each partial image. For viewpoint variation, we use GPT-Image-1.5 OpenAI (2026) to synthesize a 45-degree rotated view of the input. As shown in Figure 15, our method remains robust under severe occlusion and viewpoint changes, generating plausible 3D assets even with 75% occlusion. Quantitative results in Table 9 further confirm this robustness.

C.8 Intermediate Results for 2D Baselines

For a fair comparison with 2D controllable generation baselines, we follow a two-stage pipeline for MUSE and XVerse: each method first generates a 2D image conditioned on the input partial images and text description, and the generated image is then reconstructed into a 3D asset using TRELLIS2. In Figure 8, we show the intermediate 2D results produced by these baselines before 3D reconstruction.

Although the generated 2D images are often visually plausible, they do not always preserve all partial-image conditions faithfully. For example, some local components may be omitted, distorted, or merged into the main object appearance. Moreover, since the subsequent 3D reconstruction is performed from a single synthesized image, components that are not clearly visible in the generated

view are difficult to recover in 3D. This explains part of the performance gap between indirect 2D-to-3D baselines and our native partial-image-conditioned 3D generation framework.

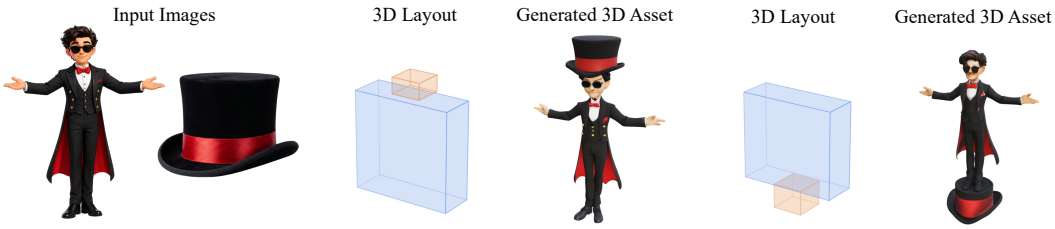

Figure 16: Resolving spatial ambiguity with 3D layouts. The same magician and hat images can correspond to multiple plausible 3D compositions. By changing only the 3D bounding-box layout, Partial3D places the hat either on the magician's head or underneath the character, demonstrating layout control over ambiguous partial-image compositions.

C.9 Layout Control for Ambiguous Compositions

Partial-image composition can be spatially ambiguous, since the same visual components may admit multiple valid 3D arrangements. As shown in Figure 16, given the same magician and hat images, different 3D bounding-box layouts lead to different generated structures. When the hat box is placed above the character, the model generates a magician wearing the hat. When the hat box is placed below the character, the model instead places the magician on top of the hat. This demonstrates that optional 3D layouts provide an effective way to disambiguate the spatial relationships among partial-image conditions.

C.10 Additional Ablation Study

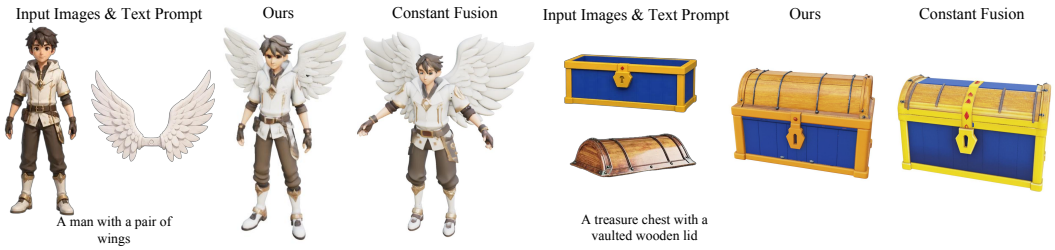

Figure 17: Ablation study of the velocity fusion strategy.

Velocity Fusion Strategy We compare our velocity fusion strategy with a constant fusion baseline, where the text-conditioned and image-conditioned velocities are combined with a fixed 50%–50% weight throughout the sparse-structure sampling process. As shown in Figure 17, constant fusion can introduce excessive text guidance in later sampling steps, which interferes with the fine-grained details provided by the image conditions. For example, the wing geometry becomes less faithful to the input wing image, and the vaulted wooden lid of the treasure chest loses some of its image-conditioned structural details. In contrast, our scheduled fusion uses text guidance mainly for early global structure formation and gradually shifts the generation toward image-conditioned details, leading to better preservation of local geometry and appearance.

User Study on Text-Controlled Fusion To evaluate the velocity-fusion schedule, we use the 34 image-and-text cases, each with its original text prompt. We compare constant text:image velocity weights of 25:75, 50:50, and 75:25 against our cosine schedule, using identical partial images, text prompts, and sampling seeds. The three constant ratios probe image-dominant, equal-weight, and text-dominant fusion, while the cosine variant uses the time-varying schedule defined in Section 3.3. The outputs are anonymized and randomly ordered, and participants rate text-structure alignment, partial-image fidelity, and compositional coherence on a 1–5 scale. An image-only variant is retained

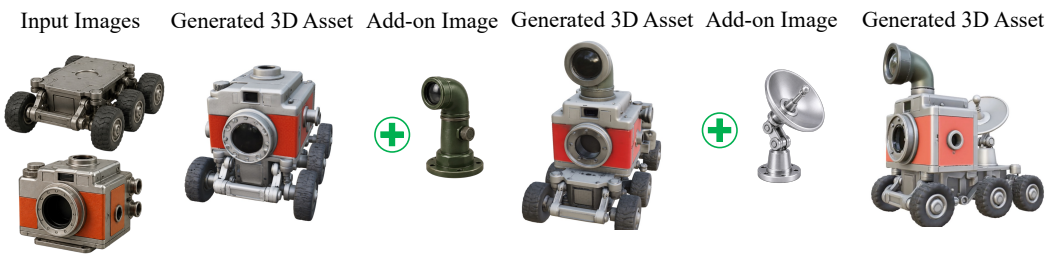

Figure 18: Incremental 3D composition with additional partial-image conditions. New components can be progressively added by introducing corresponding add-on images.

as a qualitative reference for ambiguous compositions rather than as evidence that optional text universally improves generation. Table 10 reports the resulting ratings.

Table 10: User study of velocity-fusion strategies. Ratios denote constant text:image velocity weights.

Criterion	Const. 25:75	Const. 50:50	Const. 75:25	Cosine (Ours)
Text-structure Alignment ↑	3.62	3.98	4.20	**4.23**
Partial-image Fidelity ↑	4.16	3.82	3.34	**4.21**
Compositional Coherence ↑	3.78	3.91	3.69	**4.23**

Number of Partial Images We further analyze how the number of input partial images affects generation performance. As shown in Figure 18, additional partial-image conditions can be progressively introduced, with the corresponding components incorporated into the generated 3D asset. Table 11 reports generation performance with two, three, and four partial images. These results demonstrate that our conditioning design naturally supports a variable number of visual conditions without architectural changes.

Table 11: Ablation on the number of input partial images.

Metric	2 partial images	3 partial images	4 partial images
CLIP ↑	**0.242**	**0.242**	0.233
BBox IoU ↑	0.681	**0.687**	0.678
Masked CLIP ↑	**0.574**	0.566	0.570
Masked LPIPS ↓	0.649	0.643	**0.596**
Masked DINO ↑	**0.460**	0.443	0.432

D Limitations and Future Work

Although Partial3D demonstrates strong performance in controllable 3D generation from partial images, it still has limitations. First, since our model is fine-tuned from TRELLIS2 and adopts its o-voxel representation for 3D asset generation, it naturally inherits some limitations of the backbone representation, including fixed spatial resolution and occasional small holes in the generated meshes.

Second, since TRELLIS2 does not provide a text-conditioned 3D generative model, our text conditioning relies on velocity fusion with the pretrained text-to-3D flow model from the original TRELLIS architecture. The effectiveness of text guidance is bounded by the capability of this pretrained text-conditioned model. Future work may explore directly training a unified multimodal 3D generative model in a feed-forward manner.