MoCA: Mixture-of-Components Attention for Scalable Compositional 3D Generation

Zhiqi Li[*,1,2] **Wenhuan Li**[3] **Tengfei Wang**[3,†] **Zhenwei Wang**[3] **Junta Wu**[3] **Haoyuan Wang**[3]
Yunhan Yang[3] **Zehuan Huang**[3] **Yang Li**[3] **Chunchao Guo**[3,†] **Peidong Liu**[2]
[1]Zhejiang University [2]Westlake University [3]Tencent Hunyuan



Figure 1: **MoCA** enables scalable compositional 3D generation with up to 32 components per 3D asset. It can generate part-level 3D objects with fine-grained composition, and instance-level 3D scenes with complex layouts.

Abstract

Compositionality is critical for 3D object and scene generation, but existing part-aware 3D generation methods suffer from poor scalability due to quadratic global attention costs when increasing the number of components. In this work, we present *MoCA*, a compositional 3D generative model based on a novel component-level sparse attention that explicitly models inter-component dependencies. The attention mechanism features two key designs: 1) **importance-based component routing** that utilizes a lightweight router module for cross-component importance estimation and selects top-k relevant components for fine-grained interaction, and 2) **distant components compression** that preserves spatial priors while reducing computational complexity of global attention, by including compressed distant components into attention calculation instead of discarding them. With these designs, MoCA enables intricate compositional 3D asset creation with scalable number of components. Extensive experiments show MoCA outperforms baselines on both compositional object and scene generation tasks. Project page: https://lizhiqi49.github.io/MoCA.

[*]Work done during an internship at Tencent Hunyuan; [†]Corresponding Author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Compositionality is a fundamental concept in 3D art and asset creation pipelines. It describes how complex 3D assets are formed by combining simpler, semantically coherent components, such as objects composed of simple parts and scenes composed of individual objects. These component-aware representations enable the reuse of components, targeted editing, animation, and the modeling of physically plausible interactions, all of which are essential for applications ranging from virtual production and game asset creation to robotics and computer-aided design.

Recently, 3D diffusion transformer (DiT) models [53, 23, 54, 45] have dramatically improved the quality of individual 3D object generation. However, these models commonly treat an object or scene as a monolithic entity, which limits control over the generated content and makes many downstream tasks (*e.g.*, component-level editing, per-component material customization) challenging or even unachievable. Recent works [17, 27] explore compositional 3D generation by extending pre-trained 3D shape generators [23, 18, 45] to jointly generate multiple components (either parts or instances).

While these works demonstrate that structured latent spaces enable the generation of a few semantically coherent parts and multi-object scenes, they suffer from two critical limitations when scaling up the number of components: 1) ***Scalability of global attention across components.*** To model cross-component dependencies, previous models for multi-component generation often leverage global attention modules over all component tokens by concatenating their token sets. In this naive design, N components (each represented by L latent tokens) result in global attention over $N \times L$ tokens with quadratic computational cost $O(N^2L^2)$. As N grows (*e.g.*, in complex scenes or fine-grained object decompositions), this cost quickly becomes computationally prohibitive. **2) *Uniform attention wastes capacity.*** Not all components exhibit strong interactions with one another. For example, modeling a character's hand typically only requires detailed information from the wrist and forearm, while placing a chair primarily requires information about the nearby table. Blindly enabling every token to attend to all others allocates computational resources and memory to numerous low-value interactions, leading to inefficiency and constraining model scale.

To address these problems, we propose **MoCA**, a native compositional 3D generative model equipped with a novel **M**ixture-**o**f-**C**omponents **A**ttention mechanism, designed for efficient, scalable, and accurate compositional modeling of 3D objects and scenes. Motivated by the intrinsic *sparse geometric dependencies* among 3D components, we build our method on two key designs:

- **Importance-based component routing.** Inspired by the practices of MoE (Mixture-of-Experts) models [10, 7, 19], we introduce a lightweight router module within the global block. For each component, we retain its own full-resolution tokens and select the top-$(k-1)$ other components for fine-grained interaction, giving k full-resolution components in total.

- **Compression of distant components.** We represent components not selected by the router with compressed tokens for global attention rather than discarding them. This preserves coarse-grained context (e.g., spatial priors, presence/absence cues) while dramatically reducing the number of key/value tokens in attention computations.

These designs yield a global attention context length of $L_{\text{global}} = kL + (N-k)\frac{L}{\sigma}$, where k counts full-resolution components including the query component itself, and σ is the compression ratio. With a small full-resolution fraction and a large compression ratio, this context is much shorter than that of dense global attention in prior works [17, 27], enabling efficient yet powerful compositional 3D generation.

We evaluate MoCA across both object-level and scene-level 3D generation tasks. For object-level generation, MoCA generates a 3D object from a single image as a set of automatically generated 3D parts. For scene-level generation, it produces instance-composed 3D scenes conditioned on scene images, using per-instance masks as auxiliary conditions. Extensive qualitative and quantitative experiments demonstrate that MoCA outperforms prior works by a clear margin, with particularly notable improvements in fine-grained component generation.

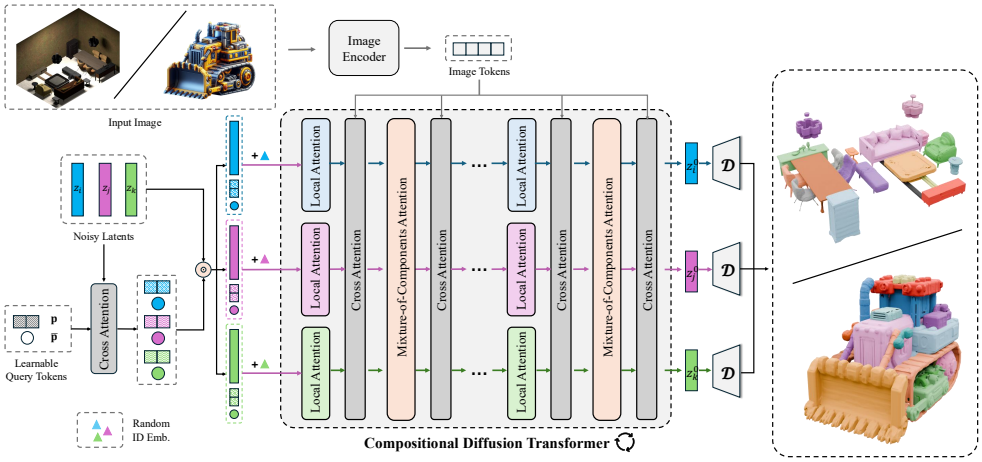

Figure 2: **Overview of MoCA**. Our DiT model starts by packing each component's latents using several learnable queries through a cross-attention layer. Random ID embeddings are applied to distinguish different components. Then, each component's full latents and compressed tokens are fed into our DiT model, which consists of interleaved local attention blocks and our proposed Mixture-of-Components Attention blocks. Finally, the clean latents of all components are separately decoded into a shared global space by a frozen shape decoder to form the final 3D asset.

2 Related Work

3D Latent Diffusion Models. Recent approaches that extend latent diffusion models (LDMs) to native 3D shape generation can be broadly categorized into two paradigms: vecset-based methods and sparse voxel-based methods. Vecset-based methods leverage a compact latent representation introduced by 3DShape2VecSet [52]. Subsequent studies [53, 22, 43, 23, 54] have advanced this paradigm to generate 3D shapes with high-resolution details, demonstrating its scalability and representational capacity. In contrast, Trellis [45] proposes a structured latent space grounded in sparse voxels. Follow-up work [50, 13, 44, 24] shows that such sparse voxel-based latent spaces excel at capturing fine-grained geometric details. Both vecset and sparse voxel-based representations produce implicit occupancy or SDF fields rather than explicit 3D meshes, so they require an iso-surface extraction step, such as Marching Cubes [32], to obtain triangular meshes.

Part-Aware 3D Object Generation. While 3D generative models have demonstrated remarkable capabilities in producing high-quality 3D objects [15, 41, 37, 42], a significant challenge lies in generating part-aware 3D objects with separate components, which are necessary for downstream tasks such as 3D editing. Early methods [29, 2] leveraged pretrained 2D priors [20, 31] to generate multi-view images with part segmentation and then reconstruct 3D geometry from them. Another line of work [48] decomposes the generation process into 3D segmentation followed by 3D part completion, but the generated parts are often difficult to assemble seamlessly. More recent native part-aware 3D generation methods [27, 38, 3, 46] typically denoise token sequences of N parts concurrently, using global attention to model their inter-part composability. However, as N grows, the quadratic increase in computational cost makes these frameworks difficult to scale. To address this limitation, we introduce a scalable solution that efficiently generates objects with more parts while significantly reducing computational overhead.

Compositional 3D Scene Generation. 3D scene generation plays a significant role in gaming and simulation, but is challenging because it requires modeling the geometry of individual objects and the complex spatial relationships between them [5, 25, 12, 39, 51]. Among these, multi-instance 3D scene generation [1, 49, 34, 33, 55, 26] has become a promising direction, focusing on the creation of multiple independent and well-arranged objects. Some methods [4, 1, 49] generate each object in a scene sequentially and then optimize their layout to compose the scene. However, this multi-stage approach is lengthy and inefficient. Recent methods [17] adopt an end-to-end paradigm, generating all 3D objects simultaneously and using global attention in diffusion transformers to model spatial relationships. However, these approaches are constrained in the number of instances they can effectively handle, typically fewer than 20. This limitation motivates our work, which focuses on enhancing the scalability of such pipelines with an efficient attention mechanism.

3 MoCA

3.1 Preliminary: Vecset Diffusion Models

Vecset diffusion models [53, 54, 23] are latent diffusion models that generate an unordered set of vectors (vecset) representing a 3D shape. The vecset VAE [52] encodes and decodes these representations through the following steps:

- Surface Sampling: Dense points $P_d \in \mathbb{R}^{N_d \times 3}$ are sampled uniformly from the shape surface or from regions near sharp edges. Sparse points $P_s \in \mathbb{R}^{N_s \times 3}$ are then obtained by downsampling P_d using farthest point sampling (FPS).
- Encoding: The sparse points P_s first aggregate features from P_d through a cross-attention layer, followed by a sequence of self-attention layers that produces the latents $\mathbf{z} \in \mathbb{R}^{N_s \times D}$:

$$\mathbf{z} = \texttt{SelfAttn}\left(\texttt{CrossAttn}\left(P_s, P_d, P_d\right)\right). \tag{1}$$

- Decoding: The decoder mirrors the encoder architecture. The latents first pass through several self-attention layers to produce features defining an implicit field. A cross-attention layer then queries the occupancy or SDF value at any spatial point $p \in \mathbb{R}^3$:

$$h = \texttt{SelfAttn}\left(\mathbf{z}\right), \hat{o} = \texttt{CrossAttn}\left(p, h, h\right). \tag{2}$$

- Surface Extraction: An explicit surface mesh is extracted from the implicit field using an iso-surface extraction method, such as Marching Cubes [32].

3.2 Model Architecture

As illustrated in Figure 2, MoCA is built on a compositional diffusion transformer (DiT) [35] with interleaved local and global attention blocks. Given an input image $\mathcal{I}$ of an object or scene and a specified component count N, our DiT jointly generates clean vecset latents $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^{N} \in \mathbb{R}^{N \times L \times D}$. A frozen vecset decoder maps each latent set to an individual 3D component: $\mathbf{c}_i = \mathcal{D}(\mathbf{z}_i)$. The decoded components together form the target 3D asset.

Our Mixture-of-Components (MoC) attention lets each component interact with selected components at full resolution and with compressed summaries of all remaining components. We construct these summaries by appending learnable query tokens to each component's noisy latents and applying cross-attention, as described in Section 3.2.1. Within each local block (Section 3.2.2), the vecset tokens and their summaries are updated using information from the same component. In each global block, a lightweight router selects which components contribute full-resolution tokens and which contribute compressed tokens for inter-component communication (Section 3.2.3).

3.2.1 Component Latents Compression

Learnable Query Tokens. Unlike image or video tokens on a regular grid, vecset latents form an unordered set. Learnable attention queries provide content-dependent summaries of these latents without requiring a fixed token ordering. We therefore append $N_p = \frac{L}{\sigma}$ learnable tokens $\mathbf{p} \in \mathbb{R}^{N_p \times D}$ to the noisy latents of each component. To support inter-component importance estimation (Section 3.2.3), we also append a learnable token $\bar{\mathbf{p}} \in \mathbb{R}^{1 \times D}$ that aggregates each component's features into a single vector. This anchor token provides the component-level representation used by the router to estimate pairwise importance scores.

Cross-Attention Packing. Before feeding tokens into the DiT model, we employ a cross-attention layer to obtain compressed representations for each component:

$$\mathbf{p}_i = \texttt{CrossAttn}(\mathbf{p}, \mathbf{z}_i, \mathbf{z}_i), \quad \bar{\mathbf{p}}_i = \texttt{CrossAttn}(\bar{\mathbf{p}}, \mathbf{z}_i, \mathbf{z}_i), \tag{3}$$

where $\mathbf{p}_i$ and $\bar{\mathbf{p}}_i$ denote the compressed tokens and anchor token of the i-th component, respectively. For clarity, we omit the diffusion timestep t in the notation.

All Input Tokens. The final input tokens for the i-th component are formed by concatenating its noisy vecset tokens $\mathbf{z}_i$, the compressed tokens $\mathbf{p}_i$, and its anchor token $\bar{\mathbf{p}}_i$: $\mathbf{x}_i = \texttt{Concat}(\mathbf{z}_i; \mathbf{p}_i; \bar{\mathbf{p}}_i)$. Following PartCrafter [27], we add random ID embeddings to distinguish components without assigning fixed semantic identities.

3.2.2 Local Feature Updating

In each local block, tokens update their features using information from the same component. To preserve the shape modeling behavior of the pretrained prior, the vecset tokens $\mathbf{z}_i$ attend only to other vecset tokens within that component and do not attend to the newly appended tokens. The compressed tokens $\mathbf{p}_i$ attend to both the vecset tokens and other compressed tokens. This allows them to refine their summaries as the vecset features evolve, while exchanging information with the other summaries of the same component. Finally, the anchor token $\bar{\mathbf{p}}_i$ attends to all tokens in $\mathbf{x}_i$, aggregating the component's full-resolution features and compressed summaries into the representation used for routing. These attention restrictions define a block-structured attention mask.

3.2.3 Mixture-of-Components Attention

During the global blocks, we perform Mixture-of-Components (MoC) attention to enable inter-component communication. Here, we describe how a component $\mathbf{c}_i$ attends to other components, as illustrated in Figure 3. The same routing and attention operations are applied to every component.

Inspired by Mixture-of-Experts (MoE) models [10, 7, 19], where router modules determine which top-k experts are selected for each token, we introduce a router module to estimate the relative importance of all other components with respect to $\mathbf{c}_i$. Based on these importance scores, the router decides whether each component should be attended through its full-token representation or through its compressed version.

Importance Score Computation. The importance score estimates a component's relevance to $\mathbf{c}_i$. We compute this score from the two components' anchor tokens using an attention-like operation at the component level.

Specifically, to compute the importance of component $\mathbf{c}_j$ with respect to $\mathbf{c}_i$, we project the anchor token of $\mathbf{c}_i$ into a query vector $\bar{Q}_i = \texttt{Query}(\bar{\mathbf{p}}_i)$, and the anchor token of $\mathbf{c}_j$ into a key vector $\bar{K}_j = \texttt{Key}(\bar{\mathbf{p}}_j)$, using two separate linear projection layers. The importance score is then given by:

$$o_{i,j} = \texttt{Sigmoid}\left(\frac{\bar{Q}_i \bar{K}_j^T}{\sqrt{d_{\bar{K}}}}\right), \qquad (4)$$

where $d_{\bar{K}}$ denotes the dimensionality of the projected key vectors. We use $\texttt{Sigmoid}(\cdot)$ rather than $\texttt{Softmax}(\cdot)$ to score each component pair independently, avoiding cross-component normalization that reduces the average gate magnitude as N grows, following a similar choice in DeepSeek-V3 [28].

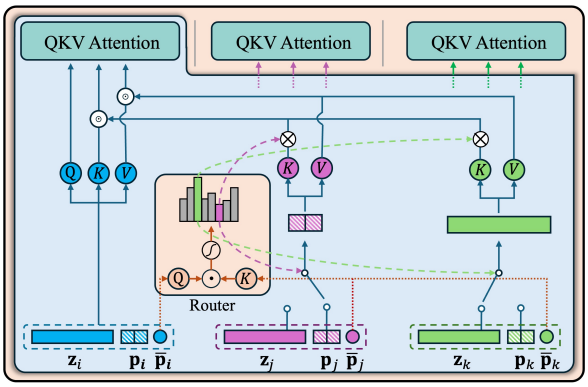

Figure 3: **Illustration of Mixture-of-Components Attention**. The computation for component $\mathbf{c}_i$ is highlighted. Selected components provide full tokens; the remaining components provide compressed tokens.

Routing Based on Importance. We count the query component itself among the k full-resolution components. At inference, we retain its own tokens and select the top-$(k-1)$ other components using $\{o_{i,j}\}_{j\neq i}$. For $j \neq i$, the routed tokens are

$$\mathbf{r}_{i,j} = \begin{cases} \mathbf{z}_j, & \text{if } o_{i,j} \in \texttt{TopK}_{k-1}\left(\{o_{i,j}\}_{j\neq i}\right), \\ \mathbf{p}_j, & \text{otherwise.} \end{cases} \qquad (5)$$

Thus, $\mathbf{c}_i$ attends to k full-resolution components, including itself, and compressed tokens from the remaining $N-k$ components.

Mixture-of-Components Attention. The context tokens that $\mathbf{c}_i$ attends to in the global block are formed by concatenating its own vecset tokens $\mathbf{z}_i$ with the routed tokens from all other components. To further modulate the contributions of different components, we apply the predicted importance



Figure 4: **Qualitative comparison for part-composed object generation**. PartPacker does not offer explicit part-count control and tends to produce coarser decompositions. In these examples, PartCrafter exhibits surface defects and floating geometry on complex objects. We use the same specified part count for PartCrafter and our model.

scores as gating factors to the key vectors of their corresponding routed tokens during attention computation. Formally, the query, key and value for $\mathbf{c}_i$ in this MoC attention are defined as:

$$\mathcal{Q}_i = \mathtt{Q}_{\mathtt{proj}}(\mathbf{x}_i), \tag{6}$$

$$\mathcal{K}_i = \mathtt{Concat}\big(\mathtt{K}_{\mathtt{proj}}(\mathbf{z}_i),\ \{o_{i,j}\mathtt{K}_{\mathtt{proj}}(\mathbf{r}_{i,j})\}_{j\neq i}\big), \tag{7}$$

$$\mathcal{V}_i = \mathtt{Concat}\big(\mathtt{V}_{\mathtt{proj}}(\mathbf{z}_i),\ \{\mathtt{V}_{\mathtt{proj}}(\mathbf{r}_{i,j})\}_{j\neq i}\big). \tag{8}$$

Multiplying the keys by the predicted importance scores modulates the corresponding attention logits before softmax. This continuous gating path allows gradients to reach the router and supports end-to-end training.

Multi-Head Routing. We apply routing independently in each attention head, allowing different heads to select different components and capture diverse inter-component dependencies, such as spatial adjacency and symmetric relationships.

Method	PartObjaverse [47]				ABO [6]			
	Self-IoU↓	**CD↓**	**Fscore-0.1↑**	**Fscore-0.05↑**	**Self-IoU↓**	**CD↓**	**Fscore-0.1↑**	**Fscore-0.05↑**
HoloPart	0.0142	0.1145	0.8340	0.6671	0.0139	0.1168	0.8523	0.6772
PartPacker	**0.0120**	0.1105	0.8484	0.6510	0.0119	0.1094	0.8646	0.6801
PartCrafter	0.0224	0.1195	0.8169	0.6236	0.0136	0.1124	0.8515	0.6759
Ours	0.0125	**0.1010**	**0.8708**	**0.6882**	**0.0116**	**0.1027**	**0.8755**	**0.6871**

Table 1: **Quantitative results for part-composed object generation**. We assess all compared methods on PartObjaverse and ABO.

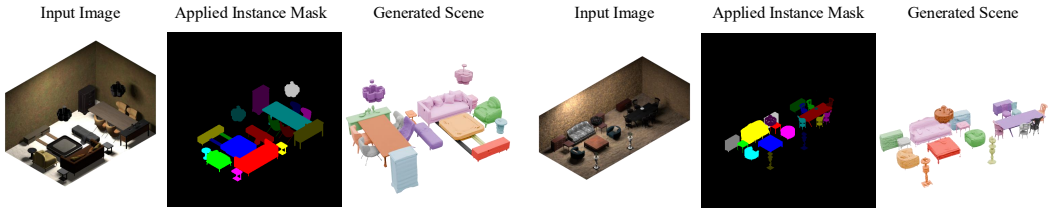

Figure 5: **Complex scene (>16 instances) generation.** MoCA jointly generates individual instances and their spatial arrangement from a scene image and per-instance masks, supporting scenes with many interacting objects.

3.3 Training

3.3.1 Load Balance Consideration

Load imbalance in MoE refers to routing collapse, where only a few experts are repeatedly selected and the remaining experts receive insufficient training [36, 7]. An analogous concern in component routing is that deterministic selection may reinforce early preferences for a small set of peers, limiting exploration of other inter-component dependencies. MoE methods commonly address load imbalance with an auxiliary loss [36, 21, 9], but its weight requires tuning and the loss may introduce undesired gradients [40]. We instead adopt an auxiliary-loss-free strategy: during training, we retain $\mathbf{c}_i$ itself and sample $k-1$ other full-resolution components from a probability distribution obtained by normalizing the importance scores $\{o_{i,j}\}_{j\neq i}$. Sampling encourages the model to explore different full-resolution interactions rather than always selecting the highest-scoring components. At inference, we use deterministic top-$(k-1)$ selection among the other components. The ablation in Table 3E evaluates the effect of this training strategy.

3.3.2 Training Objective

MoCA is trained with a rectified flow matching objective [8, 30]. Let $\mathcal{Z}_0 = \{\mathbf{z}_i^0\}_{i=1}^N \in \mathbb{R}^{N\times L\times D}$ denote the clean vecset latents of all components. At each training step, we interpolate between these latents and Gaussian noise $\epsilon \sim \mathcal{N}(0,\mathbf{I})$ at a shared noise level $t \sim \mathcal{U}(0,1)$ along the linear trajectory $\mathcal{Z}_t = (1-t)\mathcal{Z}_0 + t\epsilon$. Given $\mathcal{Z}_t$, the model predicts the velocity $\epsilon - \mathcal{Z}_0$, conditioned on the noise level t and image $\mathcal{I}$, by minimizing the following objective:

$$\mathcal{L} = \mathbb{E}_{\mathcal{Z},\epsilon,t}\left[\left\|(\epsilon - \mathcal{Z}_0) - \mathbf{v}_\theta(\mathcal{Z}_t, t, \mathcal{I})\right\|^2\right], \tag{9}$$

where $\mathbf{v}_\theta$ is the predicted velocity. During training, we drop the image condition $\mathcal{I}$ with probability 10% to enable classifier-free guidance [14].

4 Experiments

4.1 Implementation Details

Our DiT alternates between local self-attention blocks and global MoC attention blocks. In global blocks, each query component uses $k = 25\%N$ full-resolution components, including itself; the remaining components are compressed with ratio $\sigma = 8$. The random ID embedding codebook contains 50 entries. We implement routing with a custom CUDA extension for efficient tensor rearrangement. Dataset curation and training details are provided in Appendix A.1.

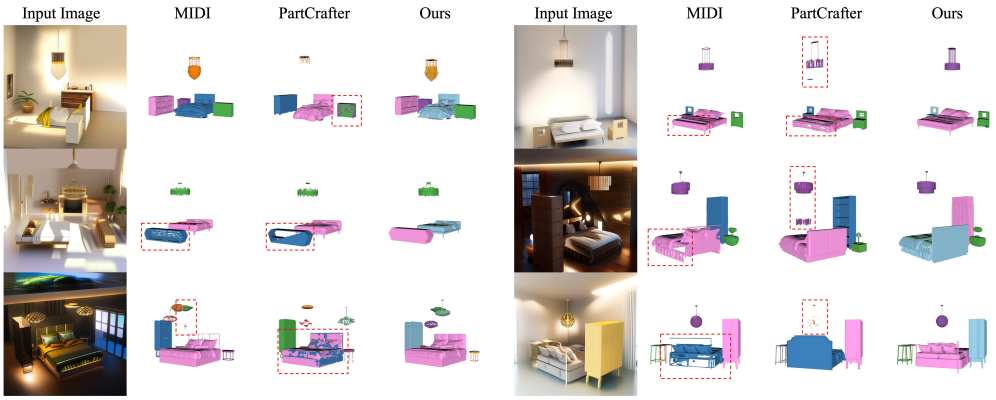

Figure 6: **Qualitative comparison on scene generation (<8 instances).** The baseline outputs exhibit surface defects in these examples. Without instance-mask conditioning, PartCrafter can confuse instance identities and generate overlapping objects.

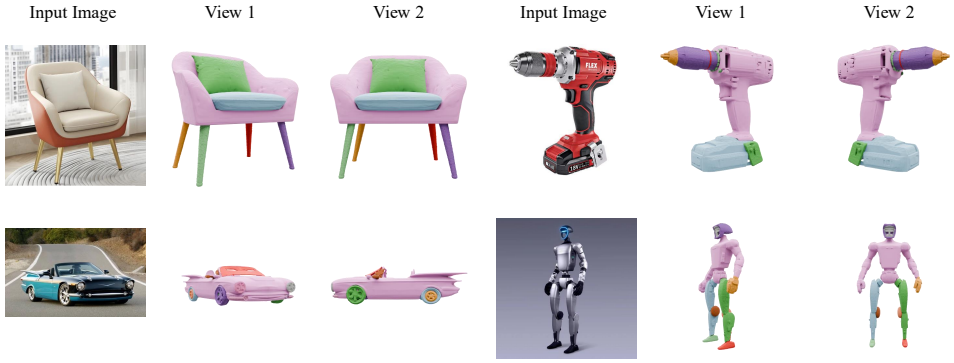

Figure 7: **Qualitative results on real-world images.** The input images were downloaded from the Internet.

4.2 Part-Composed 3D Object Generation

Evaluation Protocol. We evaluate on 200 objects from PartObjaverse [47] and 100 randomly sampled objects from ABO [6]. Generated part boundaries need not match the ground-truth decomposition, so we evaluate overall geometry using Chamfer Distance (CD) and F-score on the fused surface points of all parts. F-score thresholds of 0.1 and 0.05 capture coarse- and fine-scale geometric agreement, respectively. We additionally report self-IoU (Intersection over Union) to measure overlap between generated parts; this metric does not require correspondence with ground-truth parts.

Results. Figure 4 compares our model with PartPacker [38] and PartCrafter [27], showing finer-grained parts and cleaner decompositions in the displayed examples. We additionally evaluate HoloPart [48], using TripoSG [23] to generate the initial shape and SAMPart3D [47] to obtain its 3D segmentation. Because PartCrafter is trained exclusively on front-view images, we evaluate it on front-view renderings; all other methods use random-view renderings. The methods therefore use different input-view settings, with PartCrafter evaluated within its training distribution. In Table 1, MoCA achieves the best CD and F-scores on both datasets, with competitive inter-part overlap. Appendix A.6 reports results grouped by component count.

Generalization to Real Images We further evaluate our model qualitatively on real images. Figure 7 shows part-level decompositions of real-world objects across several categories. These examples illustrate the model's ability to generate compositional 3D objects from photographs beyond the rendered inputs used in quantitative evaluation.

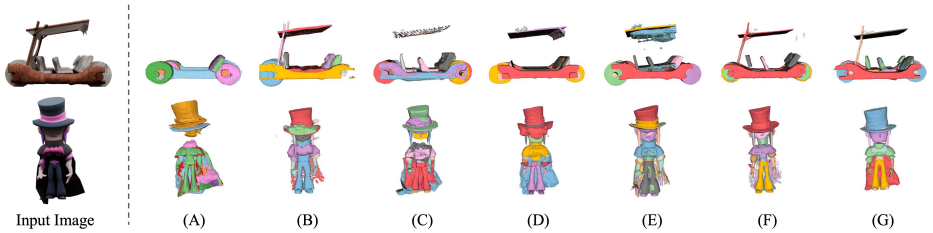

Figure 8: **Qualitative results of ablations.** Columns A–G correspond to the configurations in Table 3, with G denoting the full model.

4.3 Instance-Composed 3D Scene Generation

Evaluation Protocol. We use the 3D-FRONT [11] test split processed by MIDI [17]. Alongside self-IoU, we report scene-level and instance-level CD and F-score at threshold 0.1. MoCA and MIDI receive scene images and instance masks; PartCrafter receives only scene images. Instance-level metrics are reported for the mask-conditioned methods.

Results. We compare MoCA with MIDI [17] and PartCrafter [27]. Table 2 shows better scene-level metrics than both baselines and better instance-level metrics than MIDI. Figure 6 shows cleaner geometry and improved instance placement in the displayed examples. PartCrafter can generate duplicate or misplaced instances; its lack of instance-mask conditioning should be considered when interpreting this comparison.

Complex Scene Generation. Figure 5 shows qualitative results with >16 instances. These examples illustrate the model's ability to generate complex scenes beyond the smaller compositions shown in the baseline comparison.

Method	Self-IoU↓	CD-Scene↓	Fscore-Scene↑	CD-Obj↓	Fscore-Obj↑
MIDI	0.0011	0.1548	0.8238	0.1442	0.7996
PartCrafter	0.0036	0.1366	0.8379	-	-
Ours	**0.0006**	**0.1201**	**0.8637**	**0.1163**	**0.8380**

Table 2: **Quantitative results for scene generation**. We assess all methods on 3D-FRONT [11].

4.4 Ablations

We train each ablation variant separately on Trellis-500K [45], using the same initialization and training schedule, with up to 16 parts per object and 512 latent tokens per part. Table 3 reports results on PartObjaverse-Tiny [47]. Qualitative examples appear in Figure 8; a controlled dense-attention comparison is provided in Appendix A.5.

Necessity of routing to important components. To evaluate full-resolution interaction with selected components, we remove routing and compute global attention using only compressed features. The substantial quality drop in Table 3A shows that compressed summaries alone cannot replace detailed interactions with relevant components.

Necessity of compressed distant components. Removing the compressed tokens of unselected components substantially reduces quality (Table 3B). This supports retaining their coarse context, which preserves information about the presence and spatial arrangement of components outside the selected set.

Configuration	Routing to Important Components	Compressed Distant Components	Router Logits Multiplied to	Router Logits Activation	Load Balance	Multi-Head Routing	CD ↓	Fscore-0.1 ↑	Fscore-0.05 ↑
A	✗	✓	Key	Sigmoid	✓	✓	0.1969	0.6604	0.4750
B	✓	✗	Key	Sigmoid	✓	✓	0.1523	0.7494	0.5465
C	✓	✓	Value	Sigmoid	✓	✓	0.1519	0.7612	0.5531
D	✓	✓	Key	Softmax	✓	✓	0.1451	0.7638	0.5539
E	✓	✓	Key	Sigmoid	✗	✓	0.1361	0.7947	0.5975
F	✓	✓	Key	Sigmoid	✓	✗	0.1212	0.8215	0.6178
G (Full Model)	✓	✓	Key	Sigmoid	✓	✓	**0.1180**	**0.8259**	**0.6233**

Table 3: **Ablation study.** Each variant is trained separately on Trellis-500K [45] with the same initialization and training schedule, and evaluated on PartObjaverse-Tiny [47].

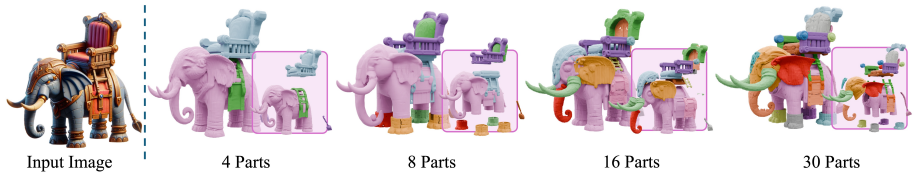

Figure 9: **Qualitative results with varying decomposition granularity.** The same input image is used with different specified part counts.

Gating keys versus values. Our model multiplies the keys by the importance scores, modulating the attention logits before softmax while retaining normalized attention weights. Applying the scores to the values instead reduces quality (Table 3C). Value gating scales contributions after softmax and changes the magnitude of the aggregated output, whereas key gating changes the attention distribution. This distinction offers a possible explanation for the observed difference; the ablation itself establishes the empirical preference for key gating.

Sigmoid versus softmax routing scores. Softmax constrains routing scores to sum to one. As the number of candidate components grows, their average magnitude decreases, which can shrink the gated keys and change the attention distribution. Sigmoid avoids this normalization, and yields better generation quality in Table 3D.

Benefits of load balance consideration. Stochastic routing encourages exploration of different component interactions during training. Replacing it with deterministic selection reduces quality (Table 3E), supporting the use of sampling during training and top-$(k-1)$ selection of other components at inference.

Effects of multi-head routing. Independent routing allows each attention head to select components relevant to the dependencies it models. Sharing a single routing decision across heads reduces quality (Table 3F), indicating the benefit of allowing different heads to use different full-resolution contexts.

Control of decomposition granularity. By specifying different component counts for the same input image, users can control the granularity of the generated asset. Figure 9 illustrates this control with 4, 8, 16, and 30 parts.

5 Conclusion

We present MoCA, a native compositional 3D generator built on Mixture-of-Components (MoC) attention. The model supports both part-composed object generation and instance-composed scene generation, improving geometric fidelity on the evaluated benchmarks while preserving separately editable components. By combining full-resolution interactions with selected components and compressed context from the remaining components, MoCA supports compositional generation with up to 32 components per asset. This design reduces the context processed by global attention while retaining information from every component.

Limitations and Future Work. We use a frozen vecset VAE, following prior work. Its reconstruction capacity can limit the quality of small components, particularly when many components occupy a shared global coordinate frame. Fine-tuning the VAE on component-level data is a promising direction for improving their reconstruction quality.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback and Dongyuan Guo for assistance with data preparation. We also thank Tencent Hunyuan for providing the computational resources used in this work.

References

[1] Andreea Ardelean, Mert Özer, and Bernhard Egger. Gen3dsr: Generalizable 3d scene reconstruction via divide and conquer from a single view. *arXiv preprint arXiv:2404.03421*, 2024.

[2] Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi. Partgen: Part-level 3d generation and reconstruction with multi-view diffusion models. *arXiv preprint arXiv:2412.18608*, 2024.

[3] Minghao Chen, Jianyuan Wang, Roman Shapovalov, Tom Monnier, Hyunyoung Jung, Dilin Wang, Rakesh Ranjan, Iro Laina, and Andrea Vedaldi. Autopartgen: Autogressive 3d part generation and discovery. *arXiv preprint arXiv:2507.13346*, 2025.

[4] Yongwei Chen, Tengfei Wang, Tong Wu, Xingang Pan, Kui Jia, and Ziwei Liu. Comboverse: Compositional 3d assets creation using spatially-aware diffusion guidance. In *European Conference on Computer Vision*, 2024.

[5] Tao Chu, Pan Zhang, Qiong Liu, and Jiaqi Wang. Buol: A bottom-up framework with occupancy-aware lifting for panoptic 3d scene reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[6] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[7] Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

[9] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[10] Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Junshi Huang. Scaling diffusion transformers to 16 billion parameters. *arXiv preprint arXiv:2407.11633*, 2024.

[11] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.

[12] Haonan Han, Rui Yang, Huan Liao, Jiankai Xing, Zunnan Xu, Xiaoming Yu, Junwei Zha, Xiu Li, and Wanhua Li. Reparo: Compositional 3d assets generation with differentiable 3d layout alignment. *arXiv preprint arXiv:2405.18525*, 2024.

[13] 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.

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

[15] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *International Conference on Learning Representations (ICLR)*, 2024.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[17] Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23646–23657, 2025.

[18] Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, et al. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025.

[19] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[20] 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*, 2023.

[21] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[22] Weiyu Li, Jiarui Liu, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024.

[23] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025.

[24] 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.

[25] Chenguo Lin and Yadong Mu. Instructscene: Instruction-driven 3d indoor scene synthesis with semantic graph prior. *arXiv preprint arXiv:2402.04717*, 2024.

[26] Chenguo Lin, Yuchen Lin, Panwang Pan, Xuanyang Zhang, and Yadong Mu. Instructlayout: Instruction-driven 2d and 3d layout synthesis with semantic graph prior. *arXiv preprint arXiv:2407.07580*, 2024.

[27] Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. *arXiv preprint arXiv:2506.05573*, 2025.

[28] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[29] Anran Liu, Cheng Lin, Yuan Liu, Xiaoxiao Long, Zhiyang Dou, Hao-Xiang Guo, Ping Luo, and Wenping Wang. Part123: part-aware 3d reconstruction from a single-view image. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

[30] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

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

[32] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.

[33] Yanxu Meng, Haoning Wu, Ya Zhang, and Weidi Xie. Scenegen: Single-image 3d scene generation in one feedforward pass. *arXiv preprint arXiv:2508.15769*, 2025.

[34] Junfeng Ni, Yu Liu, Ruijie Lu, Zirui Zhou, Song-Chun Zhu, Yixin Chen, and Siyuan Huang. Decompositional neural scene reconstruction with generative diffusion prior. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6022–6033, 2025.

[35] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[36] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[37] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision (ECCV)*, 2024.

[38] Jiaxiang Tang, Ruijie Lu, Zhaoshuo Li, Zekun Hao, Xuan Li, Fangyin Wei, Shuran Song, Gang Zeng, Ming-Yu Liu, and Tsung-Yi Lin. Efficient part-level 3d object generation via dual volume packing. *arXiv preprint arXiv:2506.09980*, 2025.

[39] HunyuanWorld Team Tencent. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels, 2025.

[40] Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024.

[41] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2023.

[42] Zhenwei Wang, Tengfei Wang, Zexin He, Gerhard Hancke, Ziwei Liu, and Rynson WH Lau. Phidias: A generative model for creating 3d content from text, image, and 3d conditions with reference-augmented diffusion. *arXiv preprint arXiv:2409.11406*, 2024.

[43] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[44] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *arXiv preprint arXiv:2505.17412*, 2025.

[45] 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. *arXiv preprint arXiv:2412.01506*, 2024.

[46] Xinhao Yan, Jiachen Xu, Yang Li, Changfeng Ma, Yunhan Yang, Chunshi Wang, Zibo Zhao, Zeqiang Lai, Yunfei Zhao, Zhuo Chen, et al. X-part: high fidelity and structure coherent shape decomposition. *arXiv preprint arXiv:2509.08643*, 2025.

[47] Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y Lam, Yan-Pei Cao, and Xihui Liu. Sampart3d: Segment any part in 3d objects. *arXiv preprint arXiv:2411.07184*, 2024.

[48] Yunhan Yang, Yuan-Chen Guo, Yukun Huang, Zi-Xin Zou, Zhipeng Yu, Yangguang Li, Yan-Pei Cao, and Xihui Liu. Holopart: Generative 3d part amodal segmentation. *arXiv preprint arXiv:2504.07943*, 2025.

[49] Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image. *ACM Transactions on Graphics (TOG)*, 44(4):1–19, 2025.

[50] 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.

[51] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T. Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, and Charles Herrmann. Wonderjourney: Going from anywhere to everywhere. In *CVPR*, 2024.

[52] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 2023.

[53] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 2024.

[54] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

[55] Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Zero-shot scene reconstruction from single images with deep prior assembly. *Advances in Neural Information Processing Systems*, 37:39104–39127, 2024.

A Appendix

A.1 Implementation Details

Backbone and Initialization. We initialize our model from a Hunyuan3D 2.0 checkpoint [54]. The backbone configuration follows Hunyuan3D 2.0, with all mixture-of-experts (MoE) layers replaced by standard feed-forward networks (FFNs), in addition to the compositional attention modifications described in Section 3.

Dataset Curation. For part-composed object generation, we train our model on an internal dataset comprising nearly 2 million watertight meshes with part-level annotations. We use objects with up to 32 parts and render images from random views for training. For instance-composed scene generation, we adapt the object model to the 3D-FRONT [11] dataset with up to 32 instances per scene. When many instances exist in a room, it is hard to capture all objects in a single perspective view. Therefore, we add randomly textured floors and walls to each scene and render images from four isometric views that clearly display all objects in the room. We mix the 3D-FRONT data processed by MIDI [17] with our newly rendered isometric images to train our scene model.

Training. We first train the initialized object model on our object dataset with 512 vecset tokens per component and a learning rate of 1e-4 for 40K steps, then finetune it with 1024 tokens per component for another 10K steps at a learning rate of 4e-5. Training uses 32 H20 GPUs with 32 components per GPU, giving a total batch size of 1024 components. Then we adapt the object model to the scene generation task by finetuning it on the scene dataset for another 10K steps with a learning rate of 4e-5 on 8 H20 GPUs. During scene task training, we use both the scene image and instance masks as conditioning, which are concatenated along the channel dimension. Therefore, following [17], we modify the number of input channels of the image encoder to 7, and finetune it using LoRA [16] along with DiT. Throughout training, we randomly drop the image condition with probability 0.1 for CFG [14]. We utilize AdamW as our optimizer, and the training precision is `BFloat16`.

Inference. Unless otherwise specified, we use 50 sampling steps and a classifier-free guidance scale of 7.0 for inference.

A.2 Additional Qualitative Results

The generated assets can be directly manipulated in 3D software such as Blender. Figure 10 illustrates editing operations including removing, moving, duplicating, and replacing components. We also provide additional qualitative results of our model on both part-level object generation and instance-level complex scene generation in Figure 11. For better visualization, we also upload **rendered videos** of more additional cases in the *supplementary materials*.

A.3 Sensitivity Analysis on the Choice of k and σ

The compression ratio $\sigma = 8$ is a conservative choice grounded in geometric intuition. In VecSet-based latent representation, a token corresponds to a surface point of the ground truth shape. Downsampling a 3D component's spatial resolution by a factor of 2 across the x, y, and z axes naturally yields a point density reduction of 1/8. This ratio effectively preserves a coarse spatial layout while safely filtering high-frequency local details.

We conducted a sensitivity analysis on the full-resolution component count k (including the query component itself, expressed as a percentage of N) under the same setting as Table 3, and the results are shown in Table 4. Performance largely stabilizes at $k \geq 20\%N$ within the tested range. Setting $k = 25\%N$ serves as a default that balances generation quality and computational efficiency.

A.4 Runtime Analysis

We conduct a runtime analysis of our method in comparison with PartCrafter [27], the most closely related baseline. Specifically, we report the inference-time breakdown of local attention, the routing procedure (MoCA only), and global attention, together with the total runtime of a single forward

k (% of N)	CD↓	Fscore-0.1↑	Fscore-0.05↑
10	0.1255	0.8084	0.5997
15	0.1222	0.8163	0.6185
20	0.1191	0.8255	0.6239
25	**0.1180**	0.8259	0.6233
30	**0.1180**	**0.8264**	**0.6241**

Table 4: **Sensitivity analysis on the choice of k.** The full-resolution count k includes the query component itself.

Number of Parts	Method	Local Attention	Routing	Global Attention	Total	Global Attention KV length
N=4	MoCA	6.9	0.97	7.5	168	1408
	PartCrafter	6.0	/	7.8	146	4096
N=8	MoCA	12	1.7	14	297	2816
	PartCrafter	12	/	18	320	8192
N=16	MoCA	25	3.3	31	583	5632
	PartCrafter	25	/	49	747	16384
N=32	MoCA	47	8.5	79	1564	11264
	PartCrafter	48	/	142	2018	32768

Table 5: **Runtime analysis of MoCA and PartCrafter [27] with different numbers of parts.** We report the runtimes of local attention, routing (MoCA only), and global attention, together with the total runtime of a single forward pass. Runtimes are in milliseconds; global attention KV lengths are in tokens.

pass and the KV length used by global attention. All experiments are performed on a single H20 GPU, with each part represented by 1024 tokens. The results are summarized in Table 5.

Thanks to the CUDA extension used to implement the routing module in global attention, the routing overhead of MoCA is now almost negligible. More importantly, MoCA substantially reduces the number of KV tokens processed by global attention. At $N = 32$, $k = 8$ includes the query component itself, giving $8 \times 1024 + 24 \times 128 = 11{,}264$ KV tokens, roughly one third of PartCrafter's 32,768. This reduction directly translates into a much lower global attention cost, decreasing the runtime from 142 ms to 79 ms at $N = 32$. As a result, while the total runtime is comparable at small part counts, once $N > 8$, MoCA is consistently faster than PartCrafter with a clear margin, and the advantage becomes more pronounced as the number of parts grows.

A.5 Comparison with Dense Global Attention

To isolate the effect of MoC attention, we compare MoCA with a variant that replaces MoC attention with dense global attention in the same architecture. Both variants are trained separately under the ablation setting of Table 3, using the same training data, initialization, number of training steps, and learning-rate schedule. We train on Trellis-500K with up to 16 components per object and $L = 512$ latent tokens per component, and evaluate on PartObjaverse-Tiny.

Global Attention	CD↓	Fscore-0.1↑	Fscore-0.05↑
Dense global attention	**0.1174**	**0.8271**	**0.6248**
MoC attention (MoCA)	0.1180	0.8259	0.6233

Table 6: **Controlled comparison with dense global attention.** Both variants use the same training setup and are evaluated on PartObjaverse-Tiny. The MoCA results correspond to row G of Table 3.

Table 6 shows that MoCA achieves geometric quality close to the dense variant: the CD differs by 0.0006, and the F-score gaps are at most 0.0015. Together with the compressed-only and top-k-only

variants in Table 3 (rows A and B), this comparison supports retaining full-resolution interactions with selected components together with compressed context from the remaining components.

A.6 Generation Quality across Component Counts

We evaluate generation quality on 300 objects from PartObjaverse and ABO, grouped by ground-truth component count, and set the generated component count to the ground-truth count. Following the object evaluation protocol in Section 4.2, Table 7 shows that MoCA achieves better geometric metrics and lower inter-component overlap than PartCrafter in every group. Both methods degrade as component counts increase, but the degradation is smaller for MoCA: from the lowest-count to the highest-count group, CD increases by 20.8% versus 60.4%, while F-score at threshold 0.1 decreases by 4.7% versus 16.9%.

GT Components	CD↓	Fscore-0.1↑	Self-IoU↓
2–4	0.0951 / 0.1032	0.8863 / 0.8571	0.0074 / 0.0103
5–8	0.0987 / 0.1121	0.8779 / 0.8332	0.0102 / 0.0178
9–16	0.1058 / 0.1338	0.8641 / 0.7861	0.0148 / 0.0312
17–32	0.1149 / 0.1655	0.8447 / 0.7124	0.0203 / 0.0521

Table 7: **Generation quality across component counts.** We group 300 objects from PartObjaverse and ABO by ground-truth component count. Each entry reports MoCA/ PartCrafter, with the generated component count set to the ground-truth count.

Impact Statement

This paper introduces MoCA, a method designed to enhance the scalability and efficiency of compositional 3D generation. The primary goal of this work is to lower the computational barriers for creating high-quality, structured 3D assets, which has positive implications for industries such as gaming, virtual reality, and computer-aided design.

However, as with other generative AI technologies, we acknowledge the potential risk of misuse, such as the generation of misleading content or the unauthorized replication of intellectual property. While our work focuses on the geometric and compositional aspects of 3D generation, we encourage the community to consider robust watermarking and attribution techniques when deploying such models in real-world applications to mitigate these risks.



Figure 10: **Qualitative results of various editing operations on our generated 3D assets.**

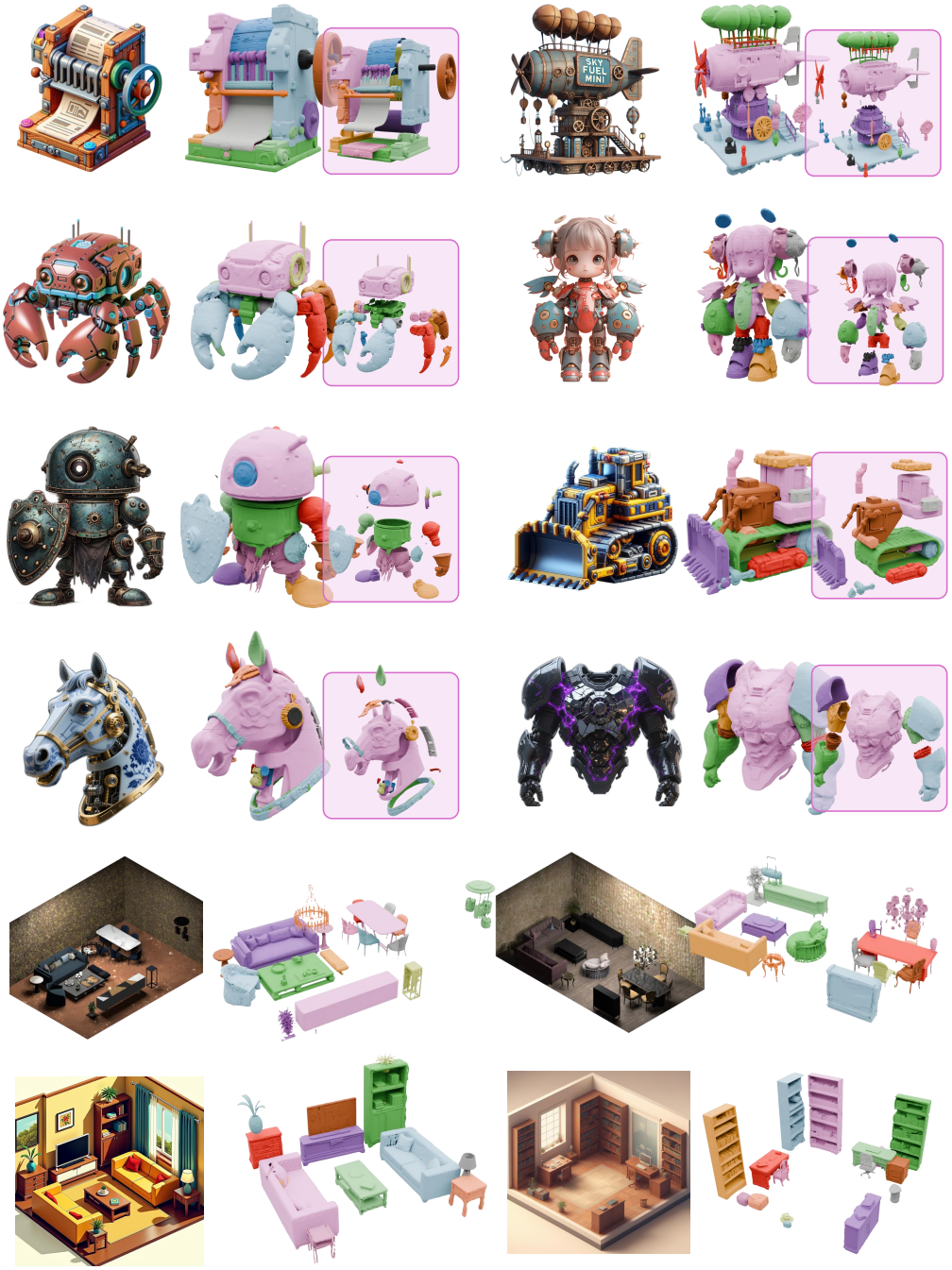

Figure 11: **Additional qualitative results of our method.**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and Introduction clearly state the proposed MoCA attention mechanism, its two main designs, and the intended scope of scalable compositional 3D object and scene generation. The claims are supported by the method description in Section 3, quantitative and qualitative results in Section 4, and runtime analysis in Appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The paper includes a "Limitations and Future Work" paragraph in Section 5. It discusses the frozen vecset VAE as a potential bottleneck when many components are generated in a shared global space.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: The paper proposes a model architecture and reports empirical results, but it does not present formal theoretical results requiring assumptions or proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No].

   Justification: Sections 3, 4, and Appendix A.1 describe the architecture, training and inference settings, evaluation protocols, and 3D-FRONT data construction. However, the object training dataset is internal and its detailed construction cannot be disclosed, limiting full reproduction of the object model and the scene model adapted from it.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: We will release our code and pretrained model weights through the project page: https://lizhiqi49.github.io/MoCA. However, the internal object training dataset cannot be publicly released, limiting full reproduction of model training.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Section 4 describes the evaluation datasets, baselines, metrics, and attention hyperparameters. Appendix A.1 specifies the Hunyuan3D 2.0 initialization and backbone configuration with MoE layers replaced by FFNs, training settings, default inference steps and guidance scale, and 3D-FRONT data construction.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

Justification: The paper does not report error bars, confidence intervals, or statistical significance tests.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No].

   Justification: Appendix A.1 reports the GPU type and counts used for object and scene training, and Table 5 reports single-forward runtime on one H20 GPU. The paper does not provide memory usage, wall-clock training time for each experiment, or a full estimate of total project compute.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes].

   Justification: The authors have reviewed the NeurIPS Code of Ethics and confirm that the research conforms to it. The paper cites the datasets, models, and baselines used for research evaluation.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: The Impact Statement in the appendix discusses potential benefits for gaming, virtual reality, and computer-aided design. It also discusses misleading content and unauthorized replication of intellectual property, and suggests watermarking and attribution as mitigation measures.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: No new datasets or model weights are publicly released with this version. Potential misuse and mitigation considerations are discussed in the Impact Statement.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: We have cited the open-source code and datasets mentioned in the paper and adhered to the terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A].

Justification: No new datasets or model weights are publicly released with this version. The project page provides release information: `https://lizhiqi49.github.io/MoCA`.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs are not used as an important, original, or non-standard component of the core method. ChatGPT was used only for grammar checking and stylistic refinement and did not affect the scientific content, experiments, analysis, or conclusions.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.