ARTUS: ARTICULATED JOINT LATENTS UNDER STRUCTURAL CONTROL FOR OBJECT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive 3D assets require part geometry, connectivity and joint motion to describe the same object across changing poses. Learning their joint distribution is difficult because compact kinematic attributes and dense geometric features play different representational roles. A joint latent representation allows kinematic structure and part geometry to interact, but leaves the direction and timing of their interaction unspecified. If geometry is mixed in early, surface variation can affect the structural path representation before joint reasoning. If geometry is kept out entirely, generated geometry cannot inform subsequent hierarchy and joint-state predictions. We propose ARTUS, a joint latent model that addresses this tension through structure-controlled joint latent refinement, where kinematic structure governs the coordination of structural and geometric information. ARTUS encodes each root-to-node structural path and uses its representation to gate geometric features before joint reasoning. The resulting joint latent supports coordinated prediction of part kinematics and geometry. To incorporate realized geometry into subsequent predictions, ARTUS introduces articulation-pathway-coupled latent refinement, where the pathway-conditioned coarse geometry latent is partially noised and then refined by a frozen text-conditioned prior. The refined latent is reintegrated into the Existing Articulation Pathway for subsequent joint reasoning. Together, these components preserve structural control through gated fusion while allowing realized geometry to progressively update the articulation pathway for later predictions. Experiments on PartNet-Mobility and PM-Openable compare ARTUS with representative baselines spanning full fusion, fully staged generation, and structure–geometry latent fusion, and dissect the design through latent-construction ablations and refinement–feedback controls across surface fidelity, articulated layout, text consistency and motion compatibility measures. Code is available at https://anonymous.4open.science/r/ARTUS-4D68.

1 INTRODUCTION

Robot simulation and virtual environments require assets that move coherently, not merely resemble objects in a single pose. A cabinet must preserve part connections while doors rotate and drawers translate. Generating such an asset requires a variable part hierarchy, local surfaces, joint axes and motion ranges to agree: a plausible closed cabinet may contain a door that intersects its frame once opened. Articulated generation therefore concerns the compatibility of geometry and motion across states, beyond static reconstruction (Xiang et al., 2020; Liu et al., 2025b).

Prior work has modeled articulated objects through structured kinematic priors, graph-based part relations and controllable joint configurations (Lei et al., 2023; Liu et al., 2024). These designs capture dependencies between shape and articulation. A central issue is ***how geometric information should enter kinematic reasoning and how its influence should evolve during generation.*** ArtFormer (Su et al., 2025) represents each articulated part with a shared token containing geometric and kinematic information, enabling their joint prediction. However, such direct coupling allows geometric features to participate in constructing kinematic representations without explicitly regulating their influence. Recent latent-based approaches, such as ArtiLatent (Chen et al., 2025), embed geometry and articulation attributes into a unified latent space for joint generation. In these formulations, structure–geometry dependencies are learned within the latent representation, but kinematic structure does not explicitly govern how geometric evidence contributes to the joint latent. By contrast, fully

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.

staged methods such as ArtLLM (Wang et al., 2026) and MeshArt (Gao et al., 2025) predict kinematic structure before synthesizing part geometry. Although MeshArt incorporates coarse latent geometry during structure generation, realized part meshes are synthesized in a subsequent stage and never reintegrated to update later structural decisions. Across these paradigms (Su et al., 2025; Chen et al., 2025; Wang et al., 2026; Gao et al., 2025), subsequent predictions therefore rely on a ***static articulation pathway*** whose context is not updated by geometry realized at preceding generation steps. For example, after a drawer is generated, its realized extent and clearance cannot inform the prediction of a nearby door or subsequent drawer, increasing the risk of inconsistent placement, motion ranges and inter-part overlap.

ARTUS addresses these two limitations through structure-controlled joint latent refinement. Within each generation step, ARTUS encodes the root-to-node structural path and uses a structure-derived controller to gate geometric features before joint prediction, explicitly regulating ***how geometry contributes to the joint latent while preserving the structural scaffold.*** Across generation steps, ARTUS replaces the static pathway with a ***dynamic articulation pathway*** by coupling geometry refinement with the Existing Articulation Pathway: the pathway-conditioned coarse latent is partially perturbed and refined by a frozen geometry prior, and the refined latent is then reintegrated for subsequent joint reasoning. Together, structure-gated interaction within a step and pathway-coupled refinement across steps form the basis of ARTUS, leading to the following contributions:

1. We propose **structure-controlled joint latent refinement** for articulated object generation, unifying structure-gated latent fusion and articulation-pathway-coupled refinement within a single framework. Kinematic structure controls how geometric evidence enters each joint latent, while newly realized geometry is reintegrated into the Existing Articulation Pathway to update subsequent joint reasoning over an evolving structure–geometry context.
2. We introduce **structure-gated latent fusion** of kinematic and geometric features. A structural path representation is combined with the local structural feature to form a structure-derived controller, which selectively gates geometric features before fusion. This prevents geometry from directly participating in structural path encoding while retaining its contribution to joint reasoning.
3. We introduce **articulation-pathway-coupled latent refinement** for autoregressive articulated generation. The coupling is realized by partially noising the pathway-conditioned coarse latent and refining it in the pretrained geometry space, thereby retaining information from the current pathway. The refined latent is then reintegrated into the Existing Articulation Pathway for subsequent joint reasoning.

2 Related Work

Part structure and articulated generation. Hierarchical shape models represented objects through related parts rather than a single surface (Du et al., 2024; Yang et al., 2022). PartNet and PartNet-Mobility supplied complementary hierarchical and motion annotations (Mo et al., 2019; Xiang et al., 2020). Graph generative priors subsequently modeled articulated attributes: NAP learned a neural articulated prior, whereas CAGE conditioned attribute generation on a category and connectivity graph (Lei et al., 2023; Liu et al., 2024). SINGAPO extended control to a single image (Liu et al., 2025a). Beyond jointly representing these attributes, ARTUS uses a path-aware articulation feature to regulate geometry before shared prediction: with the partial-tree topology and kinematic states held fixed, the structural path representation is unchanged under variations in dense geometry, while generated geometry still informs subsequent structural decisions.

Shared representations and staged generation. ArtFormer combined node attributes in a tree-autoregressive generator and used a diffusion shape prior (Su et al., 2025). Its root-to-node position encoding processed geometry with kinematic attributes, allowing shape variation to affect hierarchy representation, whereas ARTUS constructs the structural path representation from structural attributes before admitting dense geometry into shared prediction. MeshArt combined structural and mesh generation, including latent geometry in structural tokens and junction information in mesh synthesis, retaining geometric information before surface synthesis (Gao et al., 2025). ArtiLatent developed unified articulation-aware representations (Chen et al., 2025). GAOT (Sun et al., 2025), Articulate-

Anything (Le et al., 2025) and PhysX-Anything (Cao et al., 2026) explore staged pipelines or foundation models, but differ in representation and information availability.

Latent geometry and conditional fusion. Implicit shape representations and latent generative models enabled compact geometry modeling with separate surface decoders (Park et al., 2019; Yan et al., 2022; Chou et al., 2023; Xiang et al., 2025). Michelangelo further aligned shape latents with image and text features to support conditional geometry generation (Zhao et al., 2023). For feature fusion, multimodal token fusion dynamically exchanges or aggregates cross-modal features (Wang et al., 2022), while FiLM uses conditioning features to modulate intermediate activations (Perez et al., 2018). These mechanisms represent and condition geometry, but articulated generation requires deciding *who controls* the interaction: ARTUS lets a structure-derived controller—combining local structural features and the structural path representation—define the signal that gates geometric modulation before shared attention. Geometric contributions thereby adapt to the current structural state, while the structural path representation remains unchanged when the partial-tree topology and kinematic states are fixed. With latent–pathway coupling through partial-noise refinement and reintegration, this preserves a structure-defined path representation while allowing refined geometry to influence subsequent predictions.

3 METHOD

ARTUS realizes structure-controlled joint latent refinement through structure-gated latent fusion and articulation-pathway-coupled latent refinement, as summarized in Fig. 1. Sec. 3.1 defines the representation, while Secs. 3.2 and 3.3 introduce the two corresponding mechanisms.

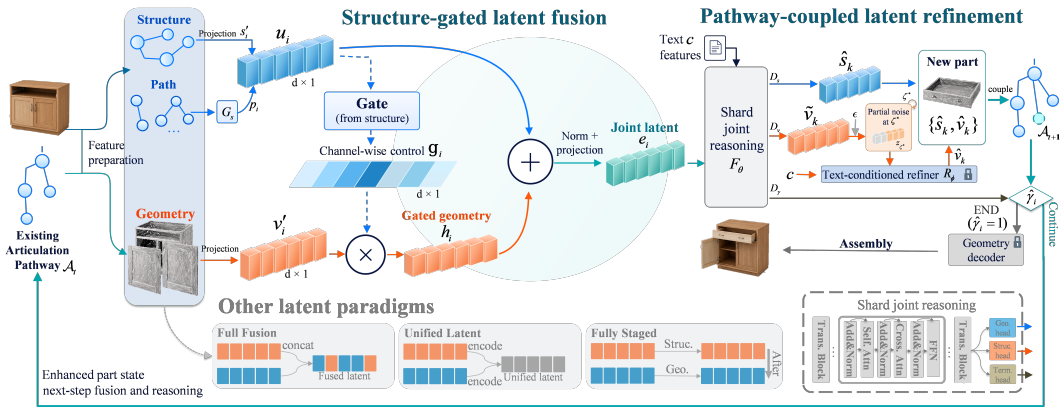

Figure 1: **Overview of ARTUS and latent interaction paradigms for articulated object generation.** ARTUS uses a structure-derived controller to gate geometric features before joint reasoning. The predicted pathway-conditioned coarse geometry latent is partially perturbed, refined by a frozen text-conditioned geometry prior, and reintegrated into the Existing Articulation Pathway for subsequent reasoning. The lower inset compares fully staged, full-fusion and unified-latent paradigms with our structure-gated formulation.

3.1 PROBLEM FORMULATION

Node state and joint representation. Given text c, the model generates a rooted tree $\tau = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the node and edge sets, and a mesh for each rigid part. For each node i, the parent index $\pi(i)$ specifies its parent–child relation. Since kinematic structure and surface geometry describe complementary aspects of a rigid part, we represent its state using explicit kinematic structure attributes for part layout and articulation, together with a dense latent code for detailed surface geometry:

$$a_i = (s_i, v_i), \qquad s_i = [b_i, o_i, d_i, l_i]. \tag{1}$$

Here, $b_i \in \mathbb{R}^6$ contains part center and extent, $o_i, d_i \in \mathbb{R}^3$ specify a joint line, and $l_i \in \mathbb{R}^4$ contains translation and rotation intervals. Together, these quantities form the structural state s_i. The geometry code v_i is the posterior-mean latent produced by a pretrained part-geometry autoencoder. During

autoregressive generation, the corresponding pathway entry is the refined realized latent $\hat{v}_i$. The root has no movable parent joint.

Origin of the part geometry latent. The geometry latent space is established before joint-generator training and remains fixed thereafter. During generation, a frozen text-conditioned latent diffusion prior R_ϕ in the geometry latent space, following the latent-diffusion formulation of (Rombach et al., 2022), refines the pathway-conditioned coarse geometry code from a selected intermediate noise level. The refined latent is used for part realization and reintegrated into the Existing Articulation Pathway for subsequent joint reasoning. The pair $(\pi(i), s_i)$ therefore specifies the hierarchical relation, part layout, and joint parameters, whereas v_i represents detailed local surface geometry. This factor boundary motivates the structure-controlled fusion in Sec. 3.2. Appendix A.1 details latent construction, pretraining and conditioning.

Existing Articulation Pathway. Let $\mathcal{V}_t$ contain all visible nodes at generation round t. The Existing Articulation Pathway $\mathcal{A}_t = \{(\pi(j), a_j) : j \in \mathcal{V}_t\}$ stores their structural and geometric states together with parent assignments. We use *pathway* for this accumulated context and *path* for an individual root-to-node structural chain. Each active parent $i \in \mathcal{V}_t$ either terminates its expansion or predicts a new child k. The joint-latent context model F_θ and factor-specific heads $D_\theta = (D_\gamma, D_s, D_v)$ predict the termination probability $\hat{\gamma}_i$, child structural state $\hat{s}_k$ and coarse geometry latent $\tilde{v}_k$:

$$(\hat{\gamma}_i, \hat{s}_k, \tilde{v}_k) = D_\theta(F_\theta(\mathcal{A}_t, c)_i), \quad \pi(k) = i. \tag{2}$$

The child state $\hat{s}_k$ and geometry latent $\tilde{v}_k$ are instantiated only when expansion is accepted. Each child attaches to an existing node, preserving an acyclic tree. All active parents in a round use the same pathway snapshot. Accepted children become visible in the next round, forming a ***dynamic articulation pathway*** whose refined geometry informs subsequent predictions. Appendix C.4 specifies expansion and termination.

3.2 Structure-Gated Latent Fusion

Factor-specific features and articulation paths. A part's articulated pose depends on the ordered joints along its ancestor chain, motivating a structural path representation beyond local attributes. For each visible part i, we project the structural state s_i and geometry latent v_i into common-width features s'_i and v'_i, respectively. The structural features along the root-to-node path induced by the parent map π are then aggregated by the structural path encoder G_s:

$$p_i = G_s\big((s'_j)_{j \in \mathrm{path}_\pi(i)}\big), \qquad u_i = s'_i + p_i. \tag{3}$$

The structural path representation p_i augments the local feature in u_i, conditioning geometric modulation on ancestor structure. Interactions with other visible parts are subsequently modeled by self-attention over joint tokens. The ordered root-to-node path is obtained by repeatedly applying the parent map π as $\mathrm{path}_\pi(i) = (\chi, \ldots, \pi^2(i), \pi(i), i)$, where χ denotes the root node. G_s applies a bidirectional GRU to the structural feature sequence along this path and projects the final bidirectional states to obtain p_i, following recent root-to-node path encodings (Su et al., 2025). The path contains only ancestors and the current node. The root uses its single-node path, and padding positions are masked. Separate mappings reflect the different statistics and semantics of the two inputs.

Structure-gated fusion. Surface evidence may affect the joint representation only after the articulation feature is established. A channel-wise gate conditioned solely on u_i, $g_i = \sigma(W_g u_i + b_g)$, modulates the geometric feature:

$$h_i = g_i \odot v'_i, \quad r = u_i + h_i, \quad e_i = W_f\,\mathrm{LN}(r) + b_f. \tag{4}$$

Here $g_i \in (0,1)^d$ is a sigmoid gate, W_g and b_g are the parameters of the gate, W_f and b_f are the parameters of the output projection, LN denotes layer normalization, and $\odot$ denotes element-wise multiplication. The gated geometry feature h_i is the geometric contribution admitted into the fusion, r is the pre-normalization and pre-projection representation, and e_i is the resulting joint latent (token) of part i. Conditioned on u_i, g_i learns which geometric channels enter e_i. Thus, geometry enters the joint representation only through h_i.

As illustrated in Fig. 2, fully staged generation limits geometry–structure interaction, the unified latent paradigm couples both factors in a common space, and full fusion directly mixes them. Our structure-gated fusion instead makes the interaction directional: u_i, formed from the local structural state and its ancestor context, determines the gate g_i, while geometry v'_i enters the joint latent only through the gated contribution $g_i \odot v'_i$. The resulting joint latent therefore retains a structure-defined baseline while admitting geometry according to the current articulation context.

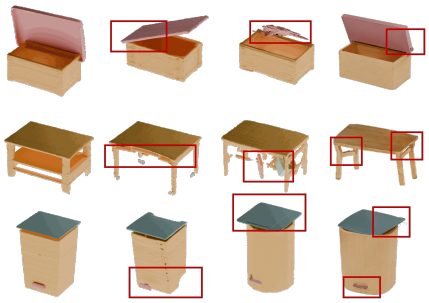

Figure 2: **Latent interaction paradigms** (left to right): our structure-gated fusion, fully staged, full fusion and unified latent.

Because g_i is recomputed from u_i for every node, the same geometry feature need not have the same effect across the hierarchy. For two parts with similar projected geometry but different structural contexts, their gated contributions can differ through g_i. The shared geometry latent is therefore interpreted according to the part's articulation role rather than being used uniformly across nodes.

Structure-governed geometry integration. Our fusion makes geometric integration explicitly governed by structure: in the representation r defined in Eq. 4, u_i provides the structural baseline, while the geometry feature v'_i contributes only through the structure-derived gate g_i, following conditional modulation (Perez et al., 2018). This separation yields an exact bound on geometric perturbations. For a geometry-latent perturbation δv_i, let $\Delta v'_i$ denote the induced change in its projected geometry feature. Since u_i and g_i depend only on structure, they remain fixed under geometry perturbations:

$$\begin{aligned} \Delta_v r &= g_i \odot \Delta v'_i, \\ \|\Delta_v r\|_2 &\le \|g_i\|_\infty \|\Delta v'_i\|_2. \end{aligned} \tag{5}$$

Since $0 < g_i < 1$ element-wise, the gate contracts geometric feature perturbations, with a separate attenuation factor for each channel. If the geometry projection is L_v-Lipschitz, Eq. 5 further gives $\|\Delta_v r\|_2 \le \|g_i\|_\infty L_v \|\delta v_i\|_2$. This bound applies before LayerNorm and output projection.

The structural contribution combines direct transmission with adaptive modulation. Define $S_i = \mathrm{diag}(g_i \odot (1 - g_i))$. Differentiating r in Eq. 4 gives

$$\frac{\partial r}{\partial v'_i} = \mathrm{diag}(g_i), \qquad \frac{\partial r}{\partial u_i} = I + \mathrm{diag}(v'_i) S_i W_g. \tag{6}$$

where I is the identity matrix. The identity term transmits structural changes directly, while the second term adjusts the contribution of geometric features. Consequently, geometry enters the joint representation through structure-dependent weights, while structure determines both the baseline representation and its sensitivity to geometric variation. Appendix A.4 details fixed-state invariance and post-projection sensitivity.

3.3 Articulation-pathway-coupled latent refinement.

Pathway-conditioned joint prediction. At generation step t, the visible joint latents in $\mathcal{A}_t$ are processed with the text condition c by F_θ as formulated in Eq. 2. As defined in Eq. 2, D_γ, D_s and D_v produce $\hat{\gamma}_i$, $\hat{s}_k$ and $\tilde{v}_k$, respectively. Each head is implemented as a lightweight stack of linear layers with nonlinear activations and a task-specific output dimension. Self-attention aggregates the accumulated structural and realized-geometry context, while cross-attention injects the frozen text condition (Vaswani et al., 2017; Raffel et al., 2020).

Pathway-coupled latent refinement. Refinement couples to the Existing Articulation Pathway through its initialization and subsequent feedback. Equation 2 conditions $\tilde{v}_k$ on $\mathcal{A}_t$ and c. The frozen refiner receives articulation context through this coarse latent, without a separate tree or joint-constraint input. We partially noise $\tilde{v}_k$ at an intermediate timestep of the geometry diffusion prior (Ho et al., 2020; Meng et al., 2022), preserving its contribution to the initialization. The current child's structural state $\hat{s}_k$ remains fixed during refinement.

Let T_{diff} denote the total number of discrete timesteps in the pretrained geometry diffusion schedule, and let $\zeta \in \{0, \dots, T_{\text{diff}} - 1\}$ index these timesteps, with larger ζ corresponding to a higher noise level. For a selected refinement starting timestep ζ^*, we initialize

$$z_{\zeta^*} = \sqrt{1 - \sigma_{\zeta^*}^2}\, \tilde{v}_k + \sigma_{\zeta^*} \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{7}$$

where σ_ζ is the noise standard deviation specified by the fixed diffusion schedule at timestep ζ. Thus, ζ^* determines the degree to which the pathway-conditioned coarse prediction is preserved before reverse refinement. For a variation δv in $\tilde{v}_k$ under the same sampled noise,

$$\|z_{\zeta^*}(\tilde{v}_k + \delta v) - z_{\zeta^*}(\tilde{v}_k)\|_2 = \sqrt{1 - \sigma_{\zeta^*}^2}\, \|\delta v\|_2. \tag{8}$$

Hence, $\sqrt{1 - \sigma_{\zeta^*}^2}$ explicitly characterizes how much variation carried by the pathway-conditioned prediction is retained at the refinement initialization. Increasing ζ^* introduces stronger noise, reducing this retention while allowing greater intervention from the pretrained geometry prior.

Starting from z_{ζ^*}, the frozen geometry prior R_ϕ (Sec. 3.1) yields $\hat{v}_k = R_\phi(z_{\zeta^*}, \zeta^*, c)$. Let $\mathcal{B}_t$ contain the children accepted in round t. Their refined states update the Existing Articulation Pathway as

$$\mathcal{A}_{t+1} = \mathcal{A}_t \cup \{(\pi(k), (\hat{s}_k, \hat{v}_k)) : k \in \mathcal{B}_t\}. \tag{9}$$

Equation 9 turns the accumulated generation history into a dynamic articulation pathway. Once a child is accepted, its refined geometry is written into $\mathcal{A}_{t+1}$ and becomes part of the context for subsequent joint prediction. The pathway therefore evolves with generation: each step augments the current articulation state with newly realized geometry, so later predictions are conditioned on the geometry that has already been formed rather than only on the coarse states available at earlier steps. This update creates a persistent coupling between refinement and autoregressive reasoning, allowing geometric information produced at one step to influence the context used at following steps while preserving the structure-defined path representation.

In the next round, $\hat{v}_k$ is projected to v'_k and enters joint reasoning through the structure-derived gate. The structural path representation p_i remains a function of the structural states along $\text{path}_\pi(i)$. Thus, refined geometry updates the pathway available to later predictions while leaving p_i unchanged for fixed topology and structural states.

We select ζ^* on the validation set and keep it fixed for all test samples. Reverse-sampling details are provided in Appendix A.2, and sensitivity to ζ^* is analyzed in Appendix D.2.

Structure-controlled joint-latent objective. The joint-latent model is trained after the geometry autoencoder and text-conditioned refiner have been pretrained and frozen. During training, teacher-forced pathways contain reference structural states and encoded geometry, whereas inference uses generated states and refined geometry codes. The joint objective supervises coarse predictions without backpropagating through refinement. The factor-specific heads are supervised by binary cross-entropy $\mathcal{L}_{\text{end}}$ for termination and normalized regression losses $\mathcal{L}_{\text{state}}$ and $\mathcal{L}_{\text{geo}}$ for the structural state and coarse geometry latent, respectively. These losses constrain the predicted factors independently, but do not explicitly supervise the structure-controlled joint representation constructed from them.

We therefore introduce an additional objective on the representation r defined in Eq. 4. For a predicted child k, $\hat{r}_k$ and r_k denote its predicted and ground-truth instances, respectively. The predicted branch re-encodes $\hat{s}_k$ and $\tilde{v}_k$, recomputing the structural path representation and gate under teacher-forced structural context. For valid parent–child pairs $\mathcal{J}$, we define

$$\mathcal{L}_{\text{scl}} = \frac{1}{|\mathcal{J}|} \sum_{(i,k) \in \mathcal{J}} \|\hat{r}_k - \text{sg}(r_k)\|_2^2, \tag{10}$$

where $\text{sg}(\cdot)$ denotes stop-gradient, its role and a symmetric control are analyzed in Appendix D.5. Unlike factor-wise regression, $\mathcal{L}_{\text{scl}}$ supervises agreement after structure-dependent fusion, including the effect of structural prediction on geometric modulation. The overall objective is

$$\mathcal{L} = \mathcal{L}_{\text{end}} + \alpha \mathcal{L}_{\text{state}} + \beta \mathcal{L}_{\text{geo}} + \lambda \mathcal{L}_{\text{scl}}. \tag{11}$$

where α, β, and λ weight the structural, geometry, and structure-controlled latent objectives, respectively. Terminal and padding positions are excluded from $\mathcal{L}_{\text{state}}$, $\mathcal{L}_{\text{geo}}$, and $\mathcal{L}_{\text{scl}}$.

4 EXPERIMENTS

We evaluate four baselines, three articulation-complexity groups, nine structure–geometry fusion controls, and five pathway-refinement controls across six metrics.

4.1 DATASET, METRICS AND IMPLEMENTATION

Dataset and coverage. PartNet-Mobility (Xiang et al., 2020) and PM-Openable (Iliash et al., 2026) supply 12 unique object categories, grouped per instance by movable-part count, a category may span several complexity groups. We retain indoor objects articulated by revolute or prismatic joints with nontrivial part hierarchies. The annotated hierarchy of each object is reconstructed and linearized by preorder traversal to form the autoregressive training sequence. Part hierarchies, meshes and joint annotations provide supervision, while rendering-derived descriptions serve as text conditions. The frozen geometry autoencoder and text-conditioned refiner are pretrained on a broader set of training categories than in joint-latent model. Appendix C.1 details data preparation and Appendix D.7 evaluates caption removal and shuffling.

Metrics and implementation. Table 1 groups the six metrics into four complementary aspects: surface fidelity via CD and F-score, articulated layout via d_{gIoU} and d_{cDist}, text consistency via CLIP based on frozen CLIP embeddings, and motion compatibility via POR. All geometric metrics are computed in the normalized object frame and, where applicable, across shared articulated states. All methods follow the normalization, matching, and motion-state sampling protocols in Appendix B. ARTUS uses a 1024-dimensional joint latent, an 8-layer, 8-head transformer and frozen 768-dimensional geometry codes. Validation-selected weights $(\alpha, \beta, \lambda) = (1, 0.5, 0.1)$ remain fixed for testing, with sensitivity analyzed in Appendix D.4. Matched variants share the frozen geometry modules and training/inference budgets specified in Appendix C.

Table 1: **Overall quality and articulation-complexity breakdown.** (a) Mean $\pm$ standard deviation over 125 test instances across 12 unique categories. (b) Subset means $\pm$ standard deviation grouped by articulation complexity. CD uses 10^{-2} units. F-score is a percentage. Best means are **bold**.

(a) Overall quality

	Surface		Layout		Text	Motion
Method	CD ↓	F-score ↑	d_{gIoU} ↓	$d_{c_{\text{Dist}}}$ ↓	CLIP ↑	POR ↓
MeshArt	$2.83_{\pm 0.54}$	$73.41_{\pm 6.34}$	$0.26_{\pm 0.07}$	$0.23_{\pm 0.06}$	$0.25_{\pm 0.05}$	$0.18_{\pm 0.06}$
MeshArt-T	$2.71_{\pm 0.48}$	$75.26_{\pm 5.72}$	$0.23_{\pm 0.06}$	$0.20_{\pm 0.05}$	$0.27_{\pm 0.04}$	$0.15_{\pm 0.05}$
ArtLLM	$2.65_{\pm 0.46}$	$75.94_{\pm 5.58}$	$0.22_{\pm 0.06}$	$0.19_{\pm 0.05}$	$0.28_{\pm 0.04}$	$0.14_{\pm 0.05}$
ArtFormer	$2.52_{\pm 0.41}$	$77.62_{\pm 4.91}$	$0.19_{\pm 0.05}$	$0.16_{\pm 0.04}$	$0.29_{\pm 0.03}$	$0.11_{\pm 0.04}$
ARTUS	$\mathbf{2.30}_{\pm 0.35}$	$\mathbf{80.30}_{\pm 4.20}$	$\mathbf{0.15}_{\pm 0.04}$	$\mathbf{0.13}_{\pm 0.03}$	$\mathbf{0.31}_{\pm 0.03}$	$\mathbf{0.08}_{\pm 0.03}$

(b) Articulation complexity

	Two-part			Three-part			Multi-part		
Method	CD ↓	d_{gIoU} ↓	POR ↓	CD ↓	d_{gIoU} ↓	POR ↓	CD ↓	d_{gIoU} ↓	POR ↓
MeshArt	$2.42_{\pm 0.35}$	$0.18_{\pm 0.04}$	$0.11_{\pm 0.03}$	$2.84_{\pm 0.51}$	$0.26_{\pm 0.07}$	$0.17_{\pm 0.05}$	$3.68_{\pm 0.74}$	$0.38_{\pm 0.10}$	$0.32_{\pm 0.09}$
MeshArt-T	$2.31_{\pm 0.31}$	$0.17_{\pm 0.04}$	$0.10_{\pm 0.03}$	$2.72_{\pm 0.46}$	$0.23_{\pm 0.06}$	$0.14_{\pm 0.05}$	$3.45_{\pm 0.68}$	$0.34_{\pm 0.09}$	$0.27_{\pm 0.08}$
ArtLLM	$2.28_{\pm 0.30}$	$0.16_{\pm 0.04}$	$0.09_{\pm 0.03}$	$2.66_{\pm 0.44}$	$0.22_{\pm 0.06}$	$0.14_{\pm 0.04}$	$3.36_{\pm 0.65}$	$0.33_{\pm 0.08}$	$0.26_{\pm 0.08}$
ArtFormer	$2.18_{\pm 0.26}$	$0.14_{\pm 0.03}$	$0.08_{\pm 0.02}$	$2.52_{\pm 0.39}$	$0.19_{\pm 0.05}$	$0.11_{\pm 0.03}$	$3.15_{\pm 0.58}$	$0.28_{\pm 0.07}$	$0.20_{\pm 0.06}$
ARTUS	$\mathbf{2.08}_{\pm 0.22}$	$\mathbf{0.13}_{\pm 0.03}$	$\mathbf{0.06}_{\pm 0.02}$	$\mathbf{2.31}_{\pm 0.31}$	$\mathbf{0.15}_{\pm 0.04}$	$\mathbf{0.08}_{\pm 0.03}$	$\mathbf{2.52}_{\pm 0.40}$	$\mathbf{0.17}_{\pm 0.04}$	$\mathbf{0.10}_{\pm 0.03}$

4.2 COMPARISON WITH EXISTING METHODS

We compare ARTUS with MeshArt (Gao et al., 2025), ArtFormer (Su et al., 2025) and ArtLLM (Wang et al., 2026), covering early-fusion, joint, and staged generation, respectively. MeshArt-T further equips MeshArt with ARTUS's frozen text encoder to control for text conditioning on the same split. As shown in Table 1(a), ARTUS achieves the best reported mean on all six metrics. Compared with ArtFormer, the strongest baseline, ARTUS reduces CD from 2.52 to 2.30, d_{gIoU} from 0.19 to 0.15,

d_{cDist} from 0.16 to 0.13, and POR from 0.11 to 0.08, while increasing F-score from 77.62 to 80.30 and CLIP from 0.29 to 0.31.

Table 1(b) further evaluates performance across articulation complexity. Errors increase with object complexity for all methods, but the advantage of ARTUS over the strongest baseline widens from two-part to complex multi-part objects on all three reported metrics. Figure A7 (Appendix D.1) illustrates the widening performance gap across complexity groups. The larger gap on multi-part objects is consistent with the intended role of structural control in coordinating geometry and articulation.

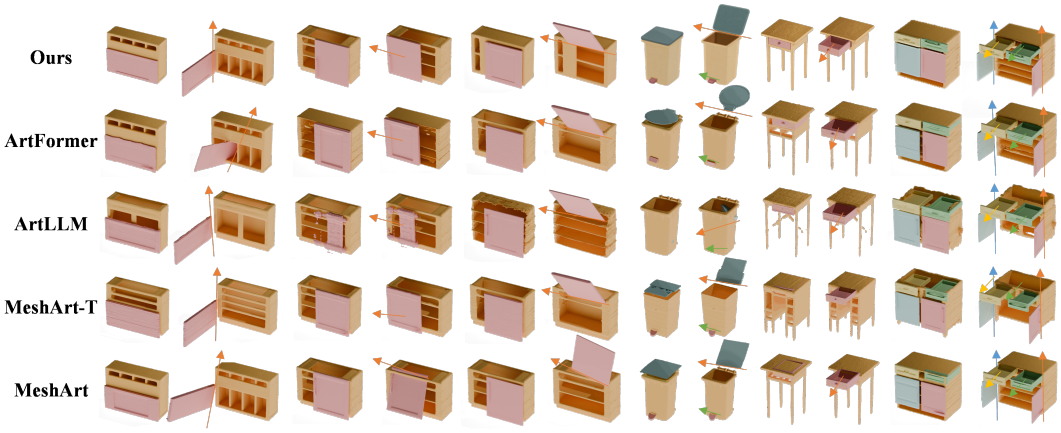

Figure 3: **Paired-state comparison.** Five methods, three examples and two poses per example. Column pairs show closed and articulated states. Colors track parts. Arrows indicate motion. ARTUS is the top row ("Ours"). Paired views reveal surface defects and part displacement that a closed pose can conceal.

Figure 3 compares generated objects in closed and articulated states. Paired states expose warped surfaces, misplaced parts, and motion inconsistencies that may remain hidden in closed poses, while ARTUS preserves more regular geometry and clearer part organization across articulation. Appendix D.8 complements this comparison with aligned-geometry diagnostics of part layout and motion compatibility across shared articulation states.

4.3 Structure-Gated Fusion Ablations

Table 2: **Structure-gated fusion ablations.** Unless otherwise specified, variants use the full root-to-node structural path. Values are mean ± standard deviation over five runs on complex objects. Metric notation and highlighting follow Table 1.

Variant	CD ↓	F-score ↑	d_{gIoU} ↓	d_{cDist} ↓	CLIP ↑	POR ↓
Local path	$3.38_{\pm 0.65}$	$65.73_{\pm 8.10}$	$0.32_{\pm 0.09}$	$0.28_{\pm 0.07}$	$0.23_{\pm 0.05}$	$0.23_{\pm 0.07}$
Constant gate	$3.04_{\pm 0.52}$	$70.28_{\pm 6.40}$	$0.27_{\pm 0.07}$	$0.23_{\pm 0.06}$	$0.26_{\pm 0.04}$	$0.18_{\pm 0.06}$
Geometry gate	$2.89_{\pm 0.46}$	$72.15_{\pm 5.60}$	$0.24_{\pm 0.06}$	$0.20_{\pm 0.05}$	$0.28_{\pm 0.04}$	$0.15_{\pm 0.05}$
Joint gate	$2.81_{\pm 0.43}$	$73.24_{\pm 5.20}$	$0.22_{\pm 0.05}$	$0.18_{\pm 0.04}$	$0.29_{\pm 0.03}$	$0.13_{\pm 0.04}$
Zero gate	$3.19_{\pm 0.58}$	$68.04_{\pm 7.30}$	$0.29_{\pm 0.08}$	$0.25_{\pm 0.06}$	$0.24_{\pm 0.05}$	$0.20_{\pm 0.06}$
Early fusion	$3.14_{\pm 0.56}$	$68.61_{\pm 7.00}$	$0.29_{\pm 0.08}$	$0.25_{\pm 0.06}$	$0.24_{\pm 0.05}$	$0.19_{\pm 0.06}$
Direct fusion	$2.96_{\pm 0.49}$	$71.03_{\pm 6.00}$	$0.26_{\pm 0.06}$	$0.22_{\pm 0.05}$	$0.27_{\pm 0.04}$	$0.16_{\pm 0.05}$
Norm. direct	$2.84_{\pm 0.45}$	$72.47_{\pm 5.40}$	$0.23_{\pm 0.05}$	$0.19_{\pm 0.05}$	$0.28_{\pm 0.04}$	$0.14_{\pm 0.05}$
ARTUS gate	$\mathbf{2.62}_{\pm 0.38}$	$\mathbf{75.61}_{\pm 4.50}$	$\mathbf{0.18}_{\pm 0.04}$	$\mathbf{0.15}_{\pm 0.04}$	$\mathbf{0.31}_{\pm 0.03}$	$\mathbf{0.10}_{\pm 0.03}$

Table 2 examines ancestor access, gate control, geometry contribution, and fusion strategy, visualized in Fig. 4. Restricting the structural path to the current node increases CD from 2.62 to 3.38 and POR from 0.10 to 0.23, showing the importance of ancestor context. With the full path fixed, the structure-derived ARTUS gate outperforms constant, geometry-derived, and joint controllers. Compared with the joint gate, it reduces CD from 2.81 to 2.62 and POR from 0.13 to 0.10. This indicates the gain comes not merely from gating, but from structure-controlled geometric modulation.

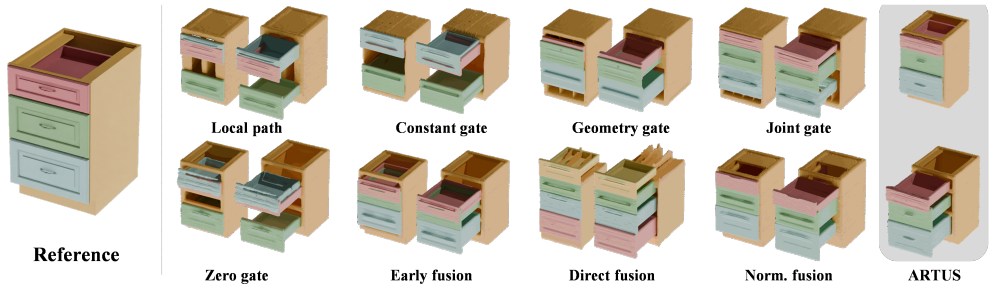

Figure 4: **Qualitative fusion ablations.** Representative outputs of Table 2. ARTUS is highlighted in gray.

Removing geometric features from joint reasoning through the zero gate substantially degrades both surface and articulation quality, confirming that geometry remains useful for joint reasoning. Early and direct fusion also underperform the proposed gate. Although normalizing direct fusion improves CD from 2.96 to 2.84 and POR from 0.16 to 0.14, ARTUS further reaches 2.62 and 0.10, showing that feature-scale normalization alone does not explain the advantage of structure-controlled fusion. Appendix D.6 reports lower mean gate activations at greater tree depths.

4.4 Pathway-Coupled Refinement Ablations

Table 3: **Pathway-coupled refinement ablations.** "Independent" starts from independent noise, "Pathway-conditioned" partially noises $\tilde{v}_k$, and "Pathway-coupled" additionally reintegrates $\hat{v}_k$ into $\mathcal{A}_{t+1}$. Values are mean $\pm$ standard deviation over five runs on complex objects. Metric notation and highlighting follow Table 1.

Variant	CD ↓	F-score ↑	d_{gIoU} ↓	$d_{c_{\text{Dist}}}$ ↓	CLIP ↑	POR ↓
Coarse only	$3.47_{\pm 0.68}$	$64.25_{\pm 8.50}$	$0.34_{\pm 0.09}$	$0.29_{\pm 0.08}$	$0.22_{\pm 0.06}$	$0.25_{\pm 0.08}$
Independent prior	$3.08_{\pm 0.53}$	$69.83_{\pm 6.50}$	$0.27_{\pm 0.07}$	$0.23_{\pm 0.06}$	$0.26_{\pm 0.05}$	$0.18_{\pm 0.06}$
Independent + feedback	$2.85_{\pm 0.45}$	$72.46_{\pm 5.50}$	$0.23_{\pm 0.06}$	$0.19_{\pm 0.05}$	$0.28_{\pm 0.04}$	$0.14_{\pm 0.05}$
Pathway-conditioned	$2.71_{\pm 0.41}$	$74.18_{\pm 4.80}$	$0.20_{\pm 0.05}$	$0.16_{\pm 0.04}$	$0.30_{\pm 0.03}$	$0.11_{\pm 0.04}$
Pathway-coupled (ARTUS)	$\mathbf{2.62}_{\pm 0.38}$	$\mathbf{75.61}_{\pm 4.50}$	$\mathbf{0.18}_{\pm 0.04}$	$\mathbf{0.15}_{\pm 0.04}$	$\mathbf{0.31}_{\pm 0.03}$	$\mathbf{0.10}_{\pm 0.03}$

Table 3 isolates refinement initialization and pathway feedback. Independent-noise refinement with the frozen geometry prior already improves over the coarse latent, reducing CD from 3.47 to 3.08 and POR from 0.25 to 0.18. Retaining the pathway-conditioned coarse prediction through partial noising improves these values further to 2.71 and 0.11, respectively, supporting refinement that preserves information already inferred from $\mathcal{A}_t$ rather than restarting independently.

Reintegrating the refined latent provides an additional gain: the pathway-conditioned variant improves from CD 2.71 and POR 0.11 to 2.62 and 0.10 with pathway feedback. The same trend also appears under independent initialization. Together, these comparisons support the two directions of the proposed coupling: the current pathway conditions geometry refinement, and the refined geometry updates the context used by subsequent predictions.

5 Conclusion

ARTUS presents a structure-controlled joint latent framework for articulated object generation, combining structure-gated fusion with articulation-pathway-coupled refinement. The former preserves a structure-defined path while selectively admitting geometric evidence, whereas the latter retains pathway-conditioned geometry during refinement and reintegrates it for subsequent reasoning. Experiments across 12 unique categories show consistent gains in surface quality, articulated layout, text alignment and motion compatibility, especially on complex multi-part objects. Ablations verify the roles of ancestor context, structure-derived gating, pathway-conditioned refinement, feedback and stop-gradient matching. Category-wise analysis shows that gains persist across varying frozen-prior quality, while efficiency analysis quantifies the added cost of pathway-coupled refinement.

REPRODUCIBILITY STATEMENT.

Implementation details, training settings, and evaluation protocols are provided in the appendices. Anonymous code is available at https://anonymous.4open.science/r/ARTUS-4D68.

AI USE STATEMENT

AI assistants contributed to this paper in two ways. First, they aided and polished the writing, including language editing, restructuring of exposition, and consistency checks of notation and terminology across sections. Second, they supported the retrieval and discovery of related work by suggesting candidate references, which the authors then located, read, and verified independently. AI assistants also helped organize tables, visualize existing numerical results, and calculate descriptive comparisons. They did not generate new experimental measurements. All assisted text, figures, and analyses require author verification before submission. The authors take full responsibility for the content of this work.

ETHICS STATEMENT

The intended application is articulated asset generation for simulation and interactive environments. Motion-aware geometric checks do not certify safety in physical deployment. Dataset terms and any redistribution permissions must be documented for the final release.

REFERENCES

Ziang Cao, Fangzhou Hong, Zhaoxi Chen, Liang Pan, and Ziwei Liu. Physx-anything: Simulation-ready physical 3d assets from single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5839–5848, 2026.

Honghua Chen, Yushi Lan, Yongwei Chen, and Xingang Pan. Artilatent: Realistic articulated 3d object generation via structured latents. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2262–2272. IEEE, 2023.

Bi'an Du, Xiang Gao, Wei Hu, and Renjie Liao. Generative 3d part assembly via part-whole-hierarchy message passing. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20850–20859. IEEE, 2024.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

Daoyi Gao, Yawar Siddiqui, Lei Li, and Angela Dai. Meshart: Generating articulated meshes with structure-guided transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 618–627. IEEE, 2025.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Denys Iliash, Hanxiao Jiang, Yiming Zhang, Manolis Savva, and Angel X Chang. S2o: Static to openable enhancement for articulated 3d objects. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6785–6795. IEEE, 2026.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. In *International Conference on Learning Representations*, volume 2025, pp. 17578–17602, 2025.

Jiahui Lei, Congyue Deng, William B Shen, Leonidas Guibas, and Kostas Daniilidis. Nap: Neural 3d articulated object prior. *Advances in Neural Information Processing Systems*, 36:31878–31894, 2023.

Jiayi Liu, Hou In Ivan Tam, Ali Mahdavi-Amiri, and Manolis Savva. Cage: Controllable articulation generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17880–17889. IEEE, 2024.

Jiayi Liu, Denys Iliash, Angel Chang, Manolis Savva, and Ali Mahdavi Amiri. Singapo: Single image controlled generation of articulated parts in objects. In *International Conference on Learning Representations*, volume 2025, pp. 97511–97532, 2025a.

Jiayi Liu, Manolis Savva, and Ali Mahdavi-Amiri. Survey on modeling of human-made articulated objects. In *Computer Graphics Forum*, volume 44, pp. e70092. Wiley Online Library, 2025b.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE.

Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 909–918. IEEE, 2019.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 165–174. IEEE, 2019.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Seyed Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666, 2019. URL https://api.semanticscholar.org/CorpusID:67855581.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Jiayi Su, Youhe Feng, Zheng Li, Jinhua Song, Yangfan He, Botao Ren, and Botian Xu. Artformer: Controllable generation of diverse 3d articulated objects. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1894–1904. IEEE, 2025.

Hao Sun, Lei Fan, Donglin Di, and Shaohui Liu. Gaot: Generating articulated objects through text-guided diffusion models. In *Proceedings of the 7th ACM International Conference on Multimedia in Asia*, pp. 1–7, 2025.

Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3400–3409. IEEE, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Penghao Wang, Siyuan Xie, Hongyu Yan, Xianghui Yang, Jingwei Huang, Chunchao Guo, and Jiayuan Gu. Artllm: Generating articulated assets via 3d llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 34281–34291, June 2026.

Yikai Wang, Xinghao Chen, Lele Cao, Wenbing Huang, Fuchun Sun, and Yunhe Wang. Multimodal token fusion for vision transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12176–12185. IEEE, 2022.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11094–11104. IEEE, 2020.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6229–6239. IEEE, 2022.

Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J Guibas, and Lin Gao. Dsg-net: Learning disentangled structure and geometry for 3d shape generation. *ACM Transactions on Graphics (TOG)*, 42(1): 1–17, 2022.

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in neural information processing systems*, 36:73969–73982, 2023.

APPENDIX

This appendix provides additional method details, evaluation protocols, implementation settings, and supplementary analyses.

- Section A specifies geometry pretraining and conditioning, the partial-noise refinement sampler, the joint-reasoning architecture and the structural-control properties of the fusion pathway.
- Section B defines the implementation of each evaluation metric, its literature basis and the aggregation and reporting units.
- Section C describes dataset composition and preprocessing, coordinate conventions, model and training settings, the generation algorithm and the baseline and control definitions.
- Section D reports supplementary analyses: articulation-complexity trends, sensitivity to the refinement starting timestep and to the loss weights, category-wise dependence on the frozen geometry prior, the stop-gradient control, gate-activation statistics, a text-conditioning ablation and aligned-geometry articulation diagnostics.

A ADDITIONAL METHOD DETAILS AND STRUCTURAL ANALYSIS

A.1 GEOMETRY REPRESENTATION, PRETRAINING AND CONDITIONING

The geometry stage establishes the latent space before joint-latent training. The geometry encoder, the text-conditioned prior, and the SDF decoder are then frozen, and all downstream components operate in this fixed latent coordinate system. Held-out test objects remain excluded from this stage. Figure A1 separates geometry representation learning from conditional prior training: the encoder first establishes the latent coordinates, and the conditional prior subsequently learns to recover a clean code from a perturbed code in that space.

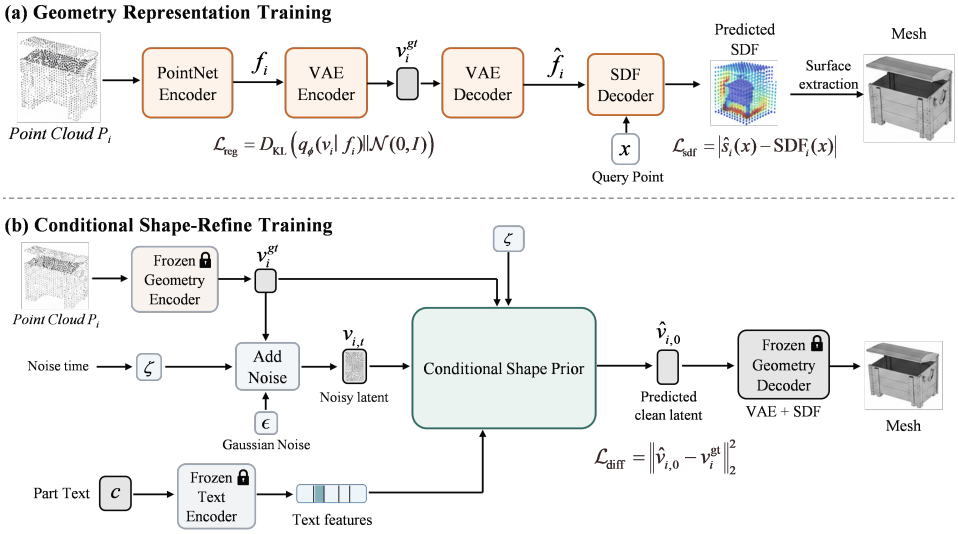

Figure A1: **Geometry representation and conditional shape-prior pretraining.** (a) Point-cloud encoding, variational latent construction, and SDF reconstruction establish the geometry space. (b) The conditional prior predicts a clean latent from its noisy counterpart, the noise timestep, and text. Locks indicate frozen modules in the corresponding stage. The clean geometry code supplies the training target.

The frozen prior and decoder share the geometry coordinates used by $\tilde{v}_k$ and $\hat{v}_k$ in Section 3.3.

A part encoder and variational autoencoder establish the geometry latent space, while an SDF decoder maps decoded features to a surface. This stage uses 4096 surface samples and 16,000 SDF queries per part, a tri-plane resolution of 64, and a 768-dimensional latent. The autoencoding objective is

$$\mathcal{L}_{\mathrm{AE}} = \mathbb{E}_x \left[|\hat{d}(x) - d(x)| \right] + \beta_{\mathrm{AE}} D_{\mathrm{KL}} \left(q_\psi(v \mid P) \,\|\, p(v) \right), \tag{12}$$

where P denotes the sampled part geometry, $d(x)$ and $\hat{d}(x)$ are the reference and reconstructed signed distances at query point x, $q_\psi(v \mid P)$ is the geometry-latent posterior, and $p(v)$ is its pretraining prior. The coefficient β_{AE} is used only for geometry pretraining and is distinct from the geometry-regression weight in the joint ARTUS objective.

The text-conditioned geometry prior is subsequently trained in this latent space. Let v_0 denote a clean geometry latent and $\zeta \in \{0, \ldots, T_{\mathrm{diff}} - 1\}$ the discrete timestep of the fixed diffusion schedule, with $T_{\mathrm{diff}} = 1000$. We parameterize the marginal noise level at timestep ζ by $\sigma_\zeta \in [0, 1]$, giving

$$v_\zeta = \sqrt{1 - \sigma_\zeta^2}\, v_0 + \sigma_\zeta \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{13}$$

Here, σ_ζ is determined by the pretrained diffusion schedule. It is not learned by ARTUS. In the clean-latent parameterization illustrated in Figure A1, a denoiser f_ϕ is supervised by

$$\mathcal{L}_{\mathrm{prior}} = \mathbb{E}_{\zeta, v_0, c, \epsilon} \left[\| f_\phi(v_\zeta, \zeta, c) - v_0 \|_2^2 \right]. \tag{14}$$

The target v_0 is a clean training code, whereas the denoiser input is v_ζ.

Conditioning interface. For the sampled geometry P_i of part i, let $q_\psi(v \mid P_i)$ denote the posterior of the pretrained geometry encoder. We use its posterior mean as the deterministic geometry target:

$$v_i = \mathbb{E}_{q_\psi(v \mid P_i)}[v] \equiv \mu_\psi(P_i). \tag{15}$$

Using the posterior mean avoids introducing an additional latent sampling variable into downstream regression.

The object-level text condition is a rendering-derived description generated from an image $I(o)$ of object o (Section C.1). Denoting the description-generation process by $\mathcal{T}$ and the frozen text encoder by E_{txt}, the conditioning feature is

$$c_o = \mathcal{T}(I(o)), \qquad h_o^c = P_c E_{\mathrm{txt}}(c_o), \tag{16}$$

where P_c projects the frozen text feature to the joint feature space. The same description conditions both joint prediction and geometry refinement.

A.2 Partial-Noise Refinement and Feedback

The pretrained geometry prior uses a diffusion schedule of $T_{\mathrm{diff}} = 1000$ discrete timesteps. At inference, the reverse process is evaluated with a 50-step sampler that retains a subset of these diffusion timesteps. The number of reverse-sampling steps and the refinement starting timestep are different quantities: the former determines the numerical sampling budget, whereas the latter determines how strongly the pathway-conditioned coarse latent is perturbed before refinement.

Let $\mathcal{S}_{\mathrm{diff}}$ denote the ordered set of diffusion timesteps retained by the 50-step sampler. We choose the refinement starting timestep $\zeta^* \in \mathcal{S}_{\mathrm{diff}}$ and initialize refinement by partially noising the coarse latent $\tilde{v}_k$ as defined in Eq. 7. When a nominal value of ζ^* does not coincide exactly with a timestep retained by the inference sampler, we use the nearest retained timestep.

Only the retained reverse updates at or below the selected starting index are traversed. At $\zeta^* = 250$, the retained schedule contains 12 reverse updates. The zero-noise endpoint is defined explicitly by $\hat{v}_k = \tilde{v}_k$, bypassing refinement.

The reverse sampler then follows the retained schedule from ζ^* toward the clean endpoint. Denoting two consecutive retained timesteps by $\zeta_j > \zeta_{j-1}$, one frozen reverse update can be written abstractly as

$$z_{\zeta_{j-1}} = \mathcal{R}_\phi \left(z_{\zeta_j}, \zeta_j, \zeta_{j-1}, c \right), \tag{17}$$

where $\mathcal{R}_\phi$ denotes the reverse update implemented by the pretrained geometry checkpoint and its fixed scheduler. Repeating these updates until the clean endpoint produces

$$\hat{v}_k = z_0. \tag{18}$$

The complete geometry-realization chain is therefore

$$\tilde{v}_k \xrightarrow{\text{partial noise at } \zeta^*} z_{\zeta^*} \xrightarrow{R_\phi(\cdot, c)} \hat{v}_k \xrightarrow{G_\psi} \mathcal{M}_k, \tag{19}$$

where G_ψ is the pretrained geometry decoder and $\mathcal{M}_k$ is the realized mesh of part k. The decoded surface is subsequently placed using the predicted part box and transformed according to the predicted kinematic hierarchy.

Finally, $\hat{v}_k$ is stored in $\mathcal{A}_{t+1}$ and becomes available to subsequent predictions.

A.3 Joint Reasoning Architecture

Figure A2 expands the joint-latent context model F_θ used in Eq. 2. At generation step t, the visible joint latents in $\mathcal{A}_t$ are processed by L shared Transformer layers. Each layer applies self-attention over the joint latents, followed by cross-attention to the projected text features and a feed-forward network. The resulting shared feature is mapped by the factor-specific heads (D_γ, D_s, D_v) to the termination probability $\hat{\gamma}_i$, structural state $\hat{s}_k$, and coarse geometry latent $\tilde{v}_k$, respectively.

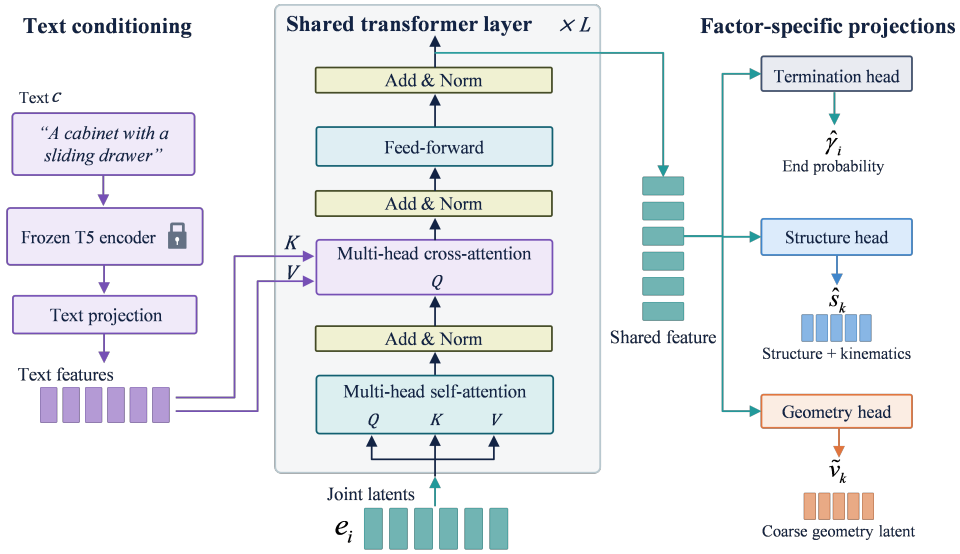

Figure A2: **Joint reasoning architecture.** Shared Transformer layers reason over joint latents under text conditioning, followed by factor-specific termination, structure, and geometry heads.

A.4 Structural-control properties

The structural path and gate invariance follow directly from the dependency graph. For fixed parent assignments and structural states, $p_i = G_s((s'_j)_{j\in \mathrm{path}_\pi(i)})$ contains no geometry input, and $g_i = \sigma(W_g(s'_i + p_i) + b_g)$ is therefore also independent of geometry. Any geometry-only transformation T consequently gives

$$p_i(s, T(v)) = p_i(s, v), \qquad g_i(s, T(v)) = g_i(s, v). \tag{20}$$

These are identities of the specified architecture.

The geometry-perturbation response of the representation r defined in Eq. 4, together with its contraction bound and Lipschitz refinement, is given in Eq. 5, while the Jacobians with respect to v'_i and u_i are given in Eq. 6. These are pre-normalization statements: the channel gate attenuates geometry-feature perturbations before normalization, while LayerNorm and the output projection also affect the sensitivity of the final token. From Eq. 4, $e_i = W_f\,\mathrm{LN}(r) + b_f$. On a region with LayerNorm Lipschitz constant L_{LN},

$$\|\Delta_v e_i\|_2 \le \|W_f\|_2 L_{\mathrm{LN}} \|g_i\|_\infty \|\Delta v'_i\|_2. \tag{21}$$

Because LayerNorm's scale depends on feature variance, a gate smaller than one does not by itself bound the final token change below the original latent perturbation.

B Evaluation Metrics and Implementation

B.1 Correspondence and measurement units

Predicted and reference parts are matched by rest-pose center distances using Hungarian assignment (Kuhn, 1955), and this correspondence is held fixed across motion states. For matched cardinalities, the assignment minimizes

$$\rho^* = \arg\min_{\rho \in \mathfrak{S}_N} \sum_{i=1}^{N} \|\hat{c}_i^{\text{rest}} - c_{\rho(i)}^{\text{rest}}\|_2, \tag{22}$$

where $\mathfrak{S}_N$ contains permutations of N reference parts. Rectangular matching applies when cardinalities differ.

Geometric distances use the normalized object frame in Section C.2. Each matched part contributes 4096 surface samples under the stated evaluation protocol. CD is displayed in units of 10^{-2}, F-score as a percentage, and the other four metrics as unscaled values.

B.2 Surface fidelity

The surface metrics use nearest-neighbor distances between sampled surfaces (Fan et al., 2017; Tatarchenko et al., 2019). For predicted and reference surface point sets P, Q, the symmetric squared Chamfer distance is

$$\text{CD}(P, Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p - q\|_2^2 + \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \|q - p\|_2^2. \tag{23}$$

Squared distance increases the influence of distant defects, yet CD can still average over localized holes or thin structures, motivating a complementary thresholded metric.

With threshold $\delta = 0.01$ of the normalized diagonal, precision, recall, and F-score are

$$P_\delta = \frac{|\{p : \min_q \|p - q\|_2 < \delta\}|}{|P|}, \quad R_\delta = \frac{|\{q : \min_p \|q - p\|_2 < \delta\}|}{|Q|}, \quad F_\delta = \frac{2P_\delta R_\delta}{P_\delta + R_\delta}. \tag{24}$$

The thresholded interpretation follows the reconstruction evaluation of Tatarchenko et al. (2019). The 1% diagonal threshold here is our normalization-specific convention.

B.3 Articulated layout

We apply the generalized-overlap construction of Rezatofighi et al. (2019) to 3D part boxes. Let $B, \hat{B}$ be reference and predicted posed part boxes, and let C be their smallest enclosing axis-aligned box. The generalized IoU distance is

$$\text{GIoU}(B, \hat{B}) = \frac{|B \cap \hat{B}|}{|B \cup \hat{B}|} - \frac{|C \setminus (B \cup \hat{B})|}{|C|}, \qquad d_{\text{gIoU}} = 1 - \text{GIoU}(B, \hat{B}). \tag{25}$$

Unlike IoU alone, this score differentiates disjoint boxes through their enclosing volume (Rezatofighi et al., 2019). It depends on box position and extent, so it complements the part-center distance

$$d_{\text{cDist}} = \|c_B - c_{\hat{B}}\|_2. \tag{26}$$

The 3D calculation replaces areas by volumes. For rotated posed boxes, the chosen box representation must be used consistently when computing overlap and enclosure.

The motion protocol uses 10 uniformly spaced fractions of each active interval, including both interval endpoints, and also evaluates the rest pose. When the rest pose is an interval endpoint, it is already included. Shared fractions are reused across methods, and the same correspondence applies throughout.

B.4 TEXT CONSISTENCY

For rendered views $\mathcal{I}_1(o), \ldots, \mathcal{I}_V(o)$ and text c, frozen CLIP encoders produce unit features

$$u_v = \frac{f_{\text{img}}(\mathcal{I}_v(o))}{\|f_{\text{img}}(\mathcal{I}_v(o))\|_2}, \qquad u_c = \frac{f_{\text{txt}}(c)}{\|f_{\text{txt}}(c)\|_2}. \tag{27}$$

The reported CLIP is the unscaled mean cosine similarity

$$\text{CLIP}(o, c) = \frac{1}{V} \sum_{v=1}^{V} u_v^\top u_c. \tag{28}$$

The frozen image/text embedding follows CLIP (Radford et al., 2021). The original captioning CLIPScore of Hessel et al. (2021) uses $2.5 \max(\cos, 0)$. Eq. 28 instead reports the unscaled cosine averaged over views.

B.5 MOTION OVERLAP

For $N \geq 2$ parts, let $\Omega_i(q)$ be part i's occupied volume at state q. The part-overlap ratio is

$$\text{POR}(q) = \frac{2}{N(N-1)} \sum_{i<j} \frac{\text{Vol}(\Omega_i(q) \cap \Omega_j(q))}{\min\{\text{Vol}(\Omega_i(q)), \text{Vol}(\Omega_j(q))\}}. \tag{29}$$

The stated protocol estimates volumes on a shared 128^3 grid and averages across motion states. With binary occupancy $O_i(x, q)$ at each grid point x, the pairwise term is implemented as

$$\widehat{\text{POR}}_{ij}(q) = \frac{\sum_{x \in \mathcal{G}} O_i(x, q) O_j(x, q)}{\min\{\sum_{x \in \mathcal{G}} O_i(x, q), \sum_{x \in \mathcal{G}} O_j(x, q)\}}, \tag{30}$$

for nonzero occupied volumes. All posed parts must be rasterized in the same object-frame grid. A zero occupied volume makes the pairwise denominator invalid and must be flagged by the evaluator. For fewer than two parts, POR is defined as zero.

Normalization by the smaller occupied volume makes a small component embedded in a larger one contribute substantially.

B.6 AGGREGATION AND SOURCES OF VARIATION

Part and motion-state averages are formed before object-level aggregation. Table 1 reports the mean and sample standard deviation across test instances, whereas Tables 2 and 3 first average the objects within each of five runs and report the mean and sample standard deviation across the run means. The supplementary tables retain the reporting unit of their source table.

Metric coverage and limitations. POR is evaluated on finitely sampled motion states and should be interpreted as an overlap metric rather than a dynamics measure.

C DATASET AND IMPLEMENTATION DETAILS

C.1 DATASET COMPOSITION AND PREPROCESSING

We use 837 PartNet-Mobility objects from 12 articulated categories (Xiang et al., 2020). PM-Openable annotations (Iliash et al., 2026) are used where available. The benchmark retains indoor objects with revolute or prismatic motion. Multiple movable parts and nontrivial hierarchies make this setting useful for evaluating whether previously generated geometry informs subsequent structural predictions. Table 4 collects the allocation reported for the manuscript, separating objects from categories.

The split is performed before geometry encoding, rendering, caption generation or partial-tree expansion, so records derived from one object never cross partition boundaries.

Table 4: **Reported benchmark allocation.** Percentages in the first block are calculated relative to 837 objects. Complexity groups are assigned per object, so the category lists in the second block overlap and together cover 12 unique categories.

Object partition	Objects	Share (%)
Optimization	670	80.04
Validation	42	5.01
Held-out test	125	14.93
Total	837	100.00
Complexity group	**Categories**	**Category names**
Two-part	10	Box, Laptop, Oven, Pliers, Refrigerator, Scissors, StorageFurniture, Table, TrashCan, USB
Three-part	4	Refrigerator, Stapler, StorageFurniture, TrashCan
Complex multi-part	4	StorageFurniture, WashingMachine, Box, Refrigerator
Unique total	12	—

Part states and hierarchy construction. Each object is represented as a rooted tree. Following Eq. 1, the node state comprises a 16-dimensional structural vector—rest-frame center and box extent (6), a point on the joint axis (3), the joint-axis direction (3), and two translation and two rotation limits (4)—together with a 768-dimensional geometry code, for 784 real-valued channels per part. A parent index and an END decision are represented separately rather than as state channels.

Preprocessing reconstructs the parent map before forming prediction targets. A preorder traversal supplies a parent-before-descendant serialization of the annotated hierarchy. This serialization is distinct from the generation schedule: Section C.4 explains how a partial tree supplies one next-child or END target for each active parent.

Geometry is encoded in a centered local part frame. Global placement remains in b_i, and joint motion remains in (o_i, d_i, l_i). This factor boundary prevents a change of surface detail from explicitly changing the numerical joint state.

Preprocessing and leakage boundaries. The construction pipeline groups duplicate objects, assigns object partitions, derives normalization statistics from optimization data, encodes part geometry, generates the text condition of each object from a rendered object image, stores parent relations, and forms partial-tree targets. Each derived record is associated with its source object, part identifier, parent identifier, local-to-object transform, text condition, and partition.

Representative articulation sequences. Figures A3–A5 visualize representative dataset objects from the three articulation-complexity groups, each shown from the rest state through intermediate motion states to the terminal articulated state. The sequence illustrates the increasing coordination required as the number of movable parts grows, from a single dominant motion in typical two-part objects to several mutually compatible joint trajectories in complex multi-part objects.

C.2 Coordinates and geometry realization

Geometric distances are expressed in a rest-pose object frame with bounding-box diagonal one. Let c_i and w_i denote the center and nonnegative full extent stored in b_i. For a canonical part point $\xi \in [-1/2, 1/2]^3$, its placement in the rest frame is

$$x_i^{\mathrm{rest}} = c_i + \mathrm{diag}(w_i)\xi. \tag{31}$$

The inverse transform uses the same box convention when encoding ground-truth geometry.

For a nonzero axis prediction, the unit direction is $\bar{d}_i = d_i/\|d_i\|_2$. A zero-length direction does not define a movable joint. For any scalar a, replacing o_i with $o_i + a\bar{d}_i$ leaves the axis line unchanged. We therefore use its nearest point to the object origin as a unique representative:

$$o_i^{\perp} = o_i - (o_i^{\top}\bar{d}_i)\bar{d}_i. \tag{32}$$

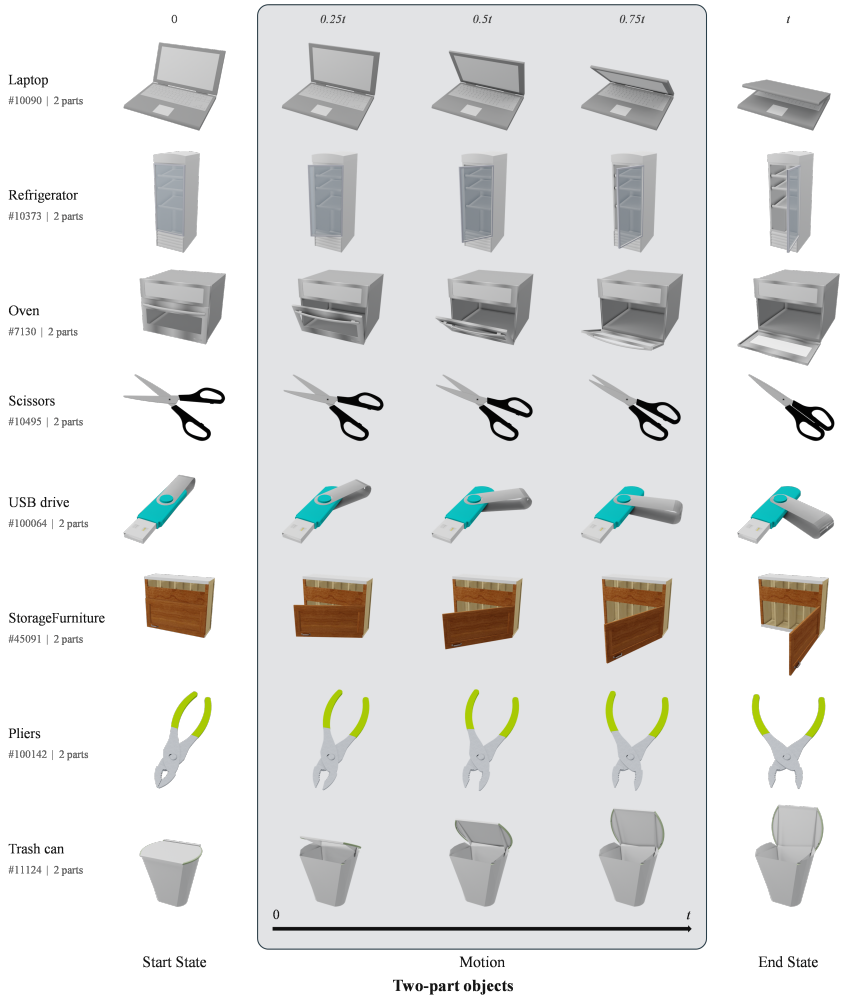

Figure A3: **Representative two-part objects.** Selected benchmark instances shown from the start state through intermediate articulation states to the end state.

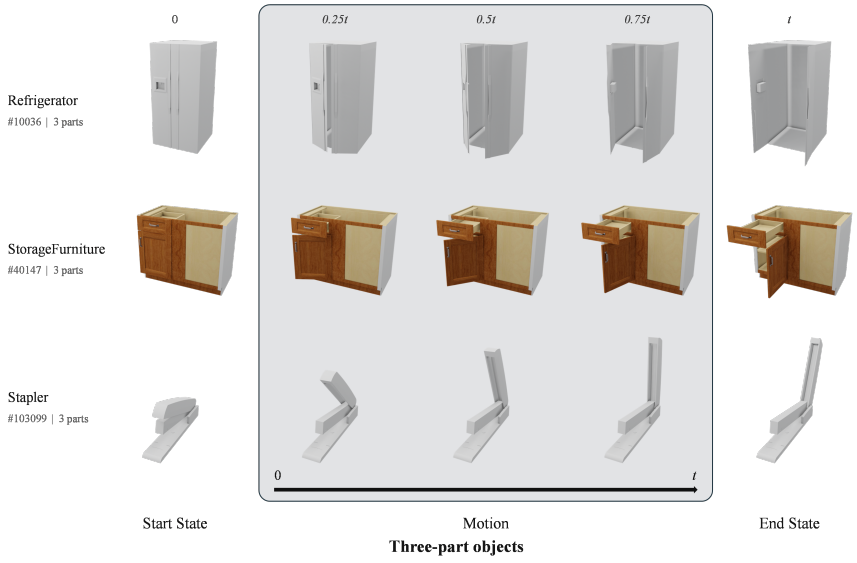

Figure A4: **Representative three-part objects.** Selected benchmark instances across their articulated motion trajectories.

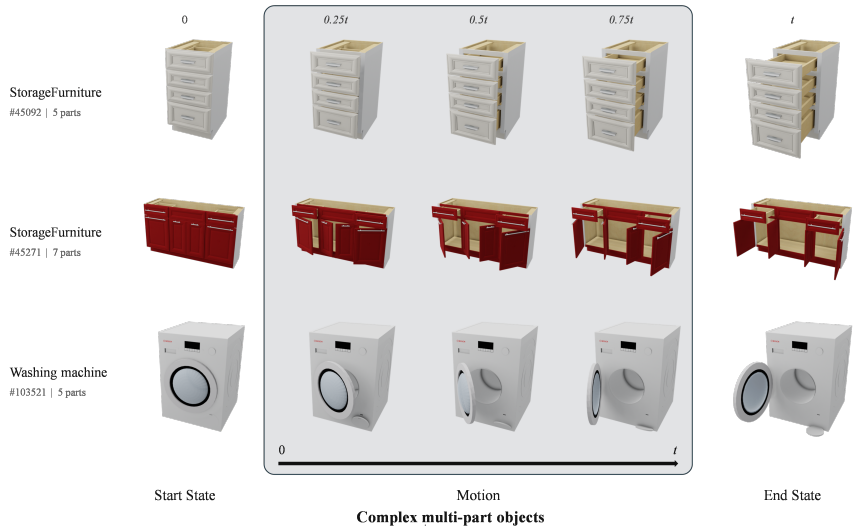

Figure A5: **Representative complex multi-part objects.** Intermediate states expose the coordinated motion of multiple articulated parts.

This removes the irrelevant displacement along the axis. Reversing the axis also preserves the same motion when the corresponding interval is transformed as $[l^-, l^+] \mapsto [-l^+, -l^-]$. Axis direction and interval endpoints must therefore be interpreted jointly.

Write $l_i = (t_i^-, t_i^+, \theta_i^-, \theta_i^+)$. A prismatic joint uses the translation interval $[t_i^-, t_i^+]$, whereas a revolute joint uses the rotation interval $[\theta_i^-, \theta_i^+]$. Angles are expressed in radians. The root has no movable parent joint. Fixed joints contribute identity transforms.

Forward kinematics. For a revolute joint, Rodrigues' formula gives the rotation about $\bar{d}_i$:

$$R_i(\theta) = \cos\theta\, I + (1 - \cos\theta)\bar{d}_i\bar{d}_i^\top + \sin\theta\, [\bar{d}_i]_\times, \tag{33}$$

where I is the 3×3 identity matrix and $[\bar{d}_i]_\times$ denotes the skew-symmetric cross-product matrix. The corresponding homogeneous transforms for revolute and prismatic motion are

$$J_i^{\text{rev}}(\theta) = \begin{bmatrix} R_i(\theta) & o_i^\perp - R_i(\theta)o_i^\perp \\ 0 & 1 \end{bmatrix}, \qquad J_i^{\text{pri}}(t) = \begin{bmatrix} I & t\bar{d}_i \\ 0 & 1 \end{bmatrix}. \tag{34}$$

Let q_j denote the active joint state of part j, and let $J_j(q_j)$ select the corresponding revolute, prismatic, or identity transform. Since the axes and rest geometry are expressed in the common rest frame, the ordered product along $\text{path}_\pi(i)$ places part i at state q:

$$\begin{bmatrix} x_i(q) \\ 1 \end{bmatrix} = \left(\prod_{j \in \text{path}_\pi(i)}^{\text{root to leaf}} J_j(q_j) \right) \begin{bmatrix} x_i^{\text{rest}} \\ 1 \end{bmatrix}. \tag{35}$$

The product order is essential because rotations and translations generally do not commute. Consequently, an error at an ancestor can displace several descendant parts even when their local meshes remain accurate.

For a normalized motion fraction $\eta \in [0, 1]$, let $[q_i^-, q_i^+]$ denote the active interval selected from l_i according to the joint type. The corresponding state is

$$q_i(\eta) = (1 - \eta)q_i^- + \eta q_i^+. \tag{36}$$

The same normalized fraction can correspond to different physical angles or translations across generated objects. Sharing η provides consistent motion-state sampling without forcing an incorrectly predicted interval to match the reference interval.

We assume an acyclic tree of single-DoF rigid joints.

C.3 Model and training specification

Table 5 consolidates the architecture and training specification of the manuscript. The structural and geometric input widths remain distinct even though both branches map to a 1024-dimensional feature space. The controller is constructed before geometric fusion, and the shared transformer reasons over the resulting joint tokens.

Table 5: **Joint-model specification.** The loss weights follow the current objective in Eqs. 10 and 11. They are distinct from geometry-pretraining regularization.

Component	Specification
Structure projection	$16 \to 4096 \to 1024$
Geometry projection	$768 \to 4096 \to 1024$
Structural path encoder	Bidirectional GRU, 512 hidden units per direction
Structural path output	Projection to 1024 channels
Gate	$1024 \to 1024$ affine map and sigmoid
Fusion	Gated sum, LayerNorm, projection to 1024
Joint transformer	8 layers, 8 attention heads, width 1024
Feed-forward width	4096
Text features	Frozen T5-large, trainable feature projection
Prediction outputs	END: 1, structure: 16, geometry: 768
Nonlinearity / dropout	Mish / 0.1
Optimizer	AdamW, learning rate 10^{-4}, weight decay 0.01
Batch size / gradient clipping	Effective batch 48 / norm 1
Joint-training budget	100,000 optimizer steps
Learning-rate schedule	5,000 warmup steps, then cosine decay
Joint-objective weights	$(\alpha, \beta, \lambda) = (1, 0.5, 0.1)$
Reported seed set	$\{0, 1, 2, 3, 4\}$

Geometry-stage pretraining settings are given in Section A.1, and the refinement sampler is specified in Section A.2. A sampler step count is distinct from the starting noise index: specifying 50 steps alone does not identify ζ^*.

Training losses and target masks. Let $\mathcal{I}$ contain active parent decisions and $\mathcal{J}$ contain accepted parent–child targets. A concrete factor-wise expression for the losses is

$$\mathcal{L}_{\text{end}} = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} [\gamma_i \log \hat{\gamma}_i + (1 - \gamma_i) \log(1 - \hat{\gamma}_i)], \tag{37}$$

$$\mathcal{L}_{\text{state}} = \frac{1}{16|\mathcal{J}|} \sum_{(i,k) \in \mathcal{J}} \|\hat{s}_k^{\text{norm}} - s_k^{\text{norm}}\|_2^2, \tag{38}$$

$$\mathcal{L}_{\text{geo}} = \frac{1}{768|\mathcal{J}|} \sum_{(i,k) \in \mathcal{J}} \|\tilde{v}_k^{\text{norm}} - v_k^{\text{norm}}\|_2^2. \tag{39}$$

Channel normalization uses optimization-set statistics with standard deviations bounded below by 10^{-6}. Normalization avoids assigning loss weight solely through raw units. The structure-controlled latent loss in Eq. 10 acts on the pre-projection fused representation and uses a stop-gradient target. A symmetric-gradient control is reported in Appendix D.5. END and padding positions have no child state or geometry target and are excluded from these regression terms.

C.4 Generation schedule and inference efficiency

The Existing Articulation Pathway stores every accepted node, including nodes that have stopped producing children. The active set contains only parents that can still expand. Each round constructs all visible tokens from the same snapshot. Children accepted within a round become visible at the next round, preventing one simultaneous prediction from using another's newly predicted attributes.

Attaching each child to an existing node preserves an acyclic tree.

Algorithm 1 Structure-controlled generation with synchronous tree expansion

Require: Text c, node cap $N_{\max}$, round cap $T_{\max}$
Predict and realize a root from the START token
Initialize pathway $\mathcal{A}$ with the root and active set $U = \{\text{root}\}$
for $t = 1, \ldots, T_{\max}$ **do**
  Snapshot $(\mathcal{A}, U)$, encode structural states and root-to-node paths
  Compute structure-derived gates, joint tokens, and shared history vectors
  Initialize accepted list $B = \emptyset$ and continuing-parent set $U' = \emptyset$
  for all active parents $i \in U$ in deterministic order **do**
    if the termination head predicts END **then**
      Retain i in $\mathcal{A}$ but retire it from expansion
    else
      if $|\mathcal{A}| + |B| = N_{\max}$ **then**
        Mark truncation and stop generation
      end if
      Predict $\hat{s}_k, \tilde{v}_k$ and assign $\pi(k) = i$
      Partially noise $\tilde{v}_k$ at ζ^* and apply the retained reverse updates of frozen R_ϕ from ζ^* to 0 to obtain $\hat{v}_k$
      Append $(\hat{s}_k, \hat{v}_k, \pi(k))$ to B, insert i into U'
    end if
  end for
  Append B to $\mathcal{A}$, set $U = U' \cup \{\text{nodes in } B\}$
  if $U = \emptyset$ **then**
    break
  end if
end for
Mark truncation if active nodes remain when the round budget is exhausted
Assemble part meshes using the stored boxes, joints, and parent map
return The articulated asset and termination/truncation status

Visual example of progressive generation. Figure A6 illustrates how successive part predictions build a four-part object.

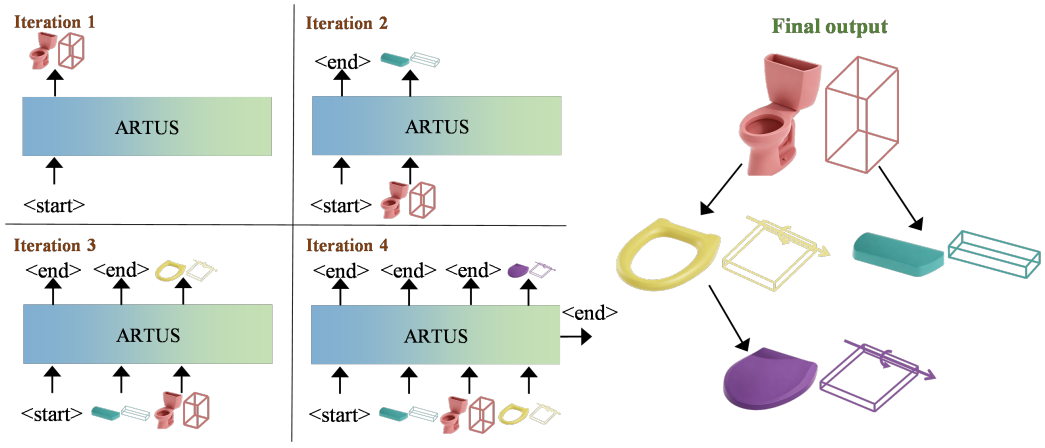

Figure A6: **Progressive articulated-part generation.** Four displayed acceptance stages accumulate the body and movable components of a toilet example. Colors associate each part geometry with its structural box, and joint arrows indicate articulated motion. Multiple parts may be accepted within one synchronous generation round.

Inference efficiency. For n_t visible nodes at round t, a dense joint-attention layer has complexity $O(n_t^2 d + n_t d^2)$. Root-to-node path encoding requires $O(n_t H_t d^2)$ up to recurrent-width constants, where H_t denotes the maximum visible hierarchy depth.

Geometry refinement introduces an additional cost for each realized part. The frozen geometry prior uses a 50-step retained inference schedule. ARTUS initializes refinement at the validation-selected $\zeta^* = 250$ and executes only the retained reverse updates from ζ^* to the clean endpoint. Let $K(\zeta^*)$ denote the corresponding number of frozen-prior evaluations per realized part. The refinement cost for an N-part object is therefore

$$C_{\text{refine}} = O\big(NK(\zeta^*)C_{R_\phi}\big)\,, \tag{40}$$

where C_{R_ϕ} denotes the cost of one reverse update of the frozen geometry prior. At $\zeta^* = 0$, refinement is bypassed and $K(0) = 0$.

To quantify practical inference cost, Table 6 reports wall-clock runtime and peak memory under a unified evaluation protocol. All methods are evaluated with batch size one on the same GPU. Runtime includes text conditioning, autoregressive structure prediction, geometry realization, and final decoding. For ARTUS, we additionally report the number of frozen-prior evaluations per realized part and the fraction of total runtime attributable to geometry refinement.

Table 6: **Inference efficiency.** Runtime is averaged per generated object under the same hardware and batch-size setting. K denotes the average number of frozen-prior evaluations per realized part. Refinement share is computed relative to the ARTUS coarse-only variant.

Method	Time/obj. (s)↓	Time/part (s)↓	K/part	Refine (%)	Mem. (GB)↓
MeshArt	38.5	14.8	–	–	6.8
ArtLLM	52.3	20.1	–	–	13.4
ArtFormer	3.8	1.5	–	–	5.2
ARTUS (coarse only)	4.2	1.6	0	0	6.9
ARTUS	**6.0**	**2.3**	**12**	**30**	**7.4**

Because refinement is applied per realized part, inference cost also varies with articulation complexity. Table 7 reports ARTUS runtime across the three complexity groups together with the average number of realized parts and frozen-prior evaluations per object.

Table 7: **ARTUS runtime across articulation complexity.** Runtime is averaged over held-out objects in each complexity group under the same inference configuration.

Complexity	Avg. parts	Time/object (s)↓	Prior eval./object
Two-part	2.0	4.9	24
Three-part	3.0	6.7	36
Complex multi-part	5.5	11.4	66

At the selected $\zeta^* = 250$, each realized part requires $K(250) = 12$ frozen-prior evaluations on average. Geometry refinement increases inference time from 4.2 s/object for the coarse-only variant to 6.0 s/object for the complete model, corresponding to a refinement share of 30% of the total ARTUS runtime.

Runtime is measured with batch size one after 20 warm-up generations and explicit CUDA synchronization, and is averaged over the held-out evaluation set. All compared methods are executed on the same NVIDIA A100 using their standard inference configurations. Peak memory is measured during the same runs.

C.5 Baseline controls and reproducibility

MeshArt retains its native conditioning, while MeshArt-T is the manuscript's text-conditioned adaptation using the frozen text encoder (not a separately named model in the original MeshArt paper). ArtFormer and ArtLLM retain their native generation designs. These comparisons measure overall method quality under the benchmark protocol, rather than isolating one module.

The internal controls instead share the split, prompts, transformer width, prediction heads, frozen geometry components, and evaluation procedure. Each configuration is trained independently.

Fusion-control definitions. The controls in Table 2 address ancestor access, gate-control source, retained geometry, and fusion scale. Table 8 specifies the changes. Here $u = h^s + p$ denotes the structural controller, h^v the encoded geometry, and a a learned vector shared by all nodes.

Table 8: **Definitions of the nine latent-construction configurations.** Every row retains geometry prediction and mesh realization. "Zero gate" removes geometry from the input to joint reasoning rather than removing the geometry output.

Configuration	Path / controller input	Geometric contribution
Local path / structure	Current node only, structural controller	Sigmoid-gated h^v
Full path / constant	Root-to-node structure, shared a	$\sigma(a) \odot h^v$
Full path / geometry	Root-to-node structure, h^v	$\sigma(W_g h^v + b_g) \odot h^v$
Full path / joint	Root-to-node structure, $(u + h^v)/\sqrt{2}$	Joint-controlled gated h^v
Full path / zero gate	Root-to-node structure	Zero contribution
Early fusion	Mixed inputs before factor encoding	Mixed structural/geometric features
Direct fusion	Root-to-node structure	Ungated h^v
Normalized direct fusion	Root-to-node structure	Separately normalized branch sum
Full path / structure	Root-to-node structure, u	$\sigma(W_g u + b_g) \odot h^v$

The local-path control runs the same recurrent path encoder on a sequence containing only the current node. This preserves nominal module capacity while removing ancestor access. The geometry, joint, and structure controllers use the same affine dimensions, enabling a narrower comparison of controller source, whereas the constant controller has fewer parameters: the affine gate in Eq. 4 contains $d^2 + d = 1{,}049{,}600$ coefficients for $d = 1024$, while a learned constant channel gate contains 1024 coefficients. Direct fusion removes the gate, and early fusion changes how factor features are formed.

For the zero-gate control, geometry features do not affect joint tokens, but the model still predicts geometry from structure and text. Both the zero-gate and full configurations preserve structural path invariance. Their quality difference therefore tests the usefulness of geometric evidence under the same local invariance property.

Refinement-control definitions. Two factors distinguish the refinement variants in Table 3: whether initialization retains the pathway-conditioned coarse code and whether the realized code is reintegrated into subsequent context. The coarse-only control decodes $\tilde{v}_k$ without refinement. The independent-prior control initializes refinement from independent noise, discarding the current coarse geometry hypothesis. Independent initialization with feedback additionally returns the refined code to subsequent joint reasoning. Pathway-conditioned refinement instead uses Eq. 7 to retain the coarse hypothesis under partial noising, but omits refined-code reintegration. Pathway-coupled ARTUS combines partial-noise initialization with the update in Eq. 9.

D Supplementary Experiments

D.1 Articulation-complexity trends

Figure A7 visualizes the group means of Table 1(b) across the three articulation-complexity groups, making the growth in surface error, layout error, and overlap directly comparable.

ARTUS has the lowest reported mean in all nine group–metric combinations, and its advantage over ArtFormer, the strongest baseline, widens with complexity for every reported metric. In absolute terms, the CD advantage grows from 0.10 to 0.21 to 0.63, the layout advantage from 0.01 to 0.04 to 0.11, and the POR advantage from 0.02 to 0.03 to 0.10.

Because the groups differ in category composition, this trend should not be attributed solely to part count.

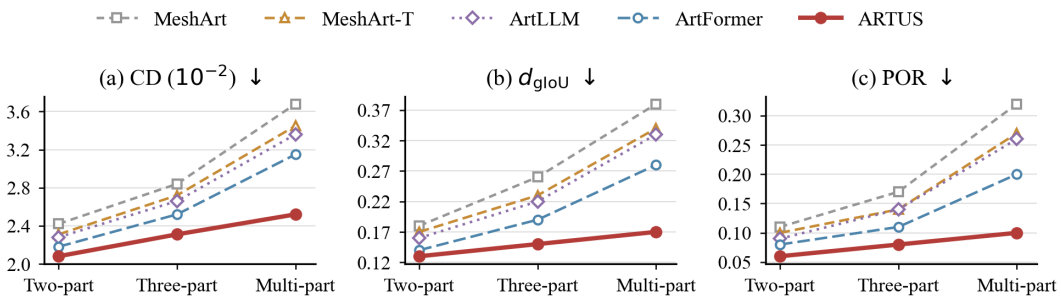

Figure A7: **Quality across articulation-complexity groups.** Curves reproduce the means in Table 1(b): CD in 10^{-2} units, d_{gIoU}, and POR. Lower is better in every panel. Lines connect categorical group means for readability. Standard deviations are reported in Table 1(b).

D.2 Sensitivity to the Refinement Starting Timestep

The starting timestep ζ^* determines the strength of partial-noise refinement and is treated as a global inference hyperparameter: it is selected on the validation set and then kept fixed for all test categories and samples, rather than predicted by an additional learned module. The sensitivity protocol considers starting timesteps distributed from weak to strong perturbation,

$$\zeta^* \in \{0, 100, 250, 400, 600, 800\}. \tag{41}$$

under the $T_{\text{diff}} = 1000$ pretrained diffusion schedule. Each nominal value is mapped to the nearest timestep retained by the 50-step inference sampler. The case $\zeta^* = 0$ serves as the no-partial-noise endpoint, while progressively larger values allow increasing intervention from the pretrained geometry prior.

Table 9 summarizes the sensitivity of geometry, layout, and motion-compatibility metrics to the refinement starting timestep ζ^*. Among the tested settings, $\zeta^* = 250$ has the best reported mean on all six metrics. Bypassing refinement ($\zeta^* = 0$) yields CD 3.47 and POR 0.25, while stronger perturbation ($\zeta^* \geq 400$) progressively increases these errors. This trend is consistent with a trade-off between retaining the coarse prediction and allowing greater intervention from the geometry prior.

Table 9: **Sensitivity to the refinement starting timestep ζ^*.** All settings evaluate complex multi-part objects with the frozen geometry prior, 50-step inference sampler, and pathway feedback. The validation-selected setting ($\zeta^* = 250$) is highlighted in **bold**.

ζ^*	CD ↓	F-score ↑	d_{gIoU} ↓	d_{cDist} ↓	CLIP ↑	POR ↓
0	3.47	64.25	0.34	0.29	0.22	0.25
100	2.92	71.38	0.24	0.20	0.27	0.15
250	**2.62**	**75.61**	**0.18**	**0.15**	**0.31**	**0.10**
400	2.74	74.12	0.20	0.17	0.30	0.12
600	2.89	72.05	0.23	0.19	0.29	0.15
800	3.05	70.18	0.26	0.22	0.27	0.17

D.3 Category-wise dependence on the frozen geometry prior

The geometry autoencoder and text-conditioned refinement prior are pretrained before joint-latent learning and remain frozen throughout ARTUS training. We examine whether category-wise variation in their reconstruction and recovery quality is associated with the relative improvement achieved by ARTUS.

Category-wise geometry quality. For each of the 12 semantic categories represented in the held-out partition, let $\mathcal{O}_c$ denote the set of test objects belonging to category c, and let $\mathcal{P}_o$ denote the rigid parts of object o. Following Eq. 15, the clean latent of part i is $v_i = \mu_\psi(P_i)$.

We first evaluate the reconstruction quality of the fixed geometry representation by decoding the clean posterior-mean latent:

$$Q_{\mathrm{AE}}(c) = \frac{1}{|\mathcal{O}_c|} \sum_{o \in \mathcal{O}_c} \frac{1}{|\mathcal{P}_o|} \sum_{i \in \mathcal{P}_o} \mathrm{CD}\left(G_\psi(v_i), P_i\right). \tag{42}$$

Here, G_ψ is the frozen geometry decoder defined in Eq. 19. Lower Q_{AE} indicates more accurate reconstruction by the pretrained latent representation and decoder. CD follows the same normalized-coordinate, surface-sampling, and 10^{-2} reporting convention as Section B, and averaging first over parts within each object gives every object equal weight.

Autoencoder reconstruction does not directly characterize the behavior of the frozen text-conditioned prior. We therefore evaluate controlled prior recovery using the same refinement starting timestep as the main ARTUS configuration. For every clean latent v_i, we construct

$$z_{\zeta^*}^{(i,m)} = \sqrt{1 - \sigma_{\zeta^*}^2}\, v_i + \sigma_{\zeta^*} \epsilon^{(m)}, \qquad \epsilon^{(m)} \sim \mathcal{N}(0, I), \tag{43}$$

with $\zeta^* = 250$. Starting from $z_{\zeta^*}^{(i,m)}$, we run the same frozen reverse sampler described in Section A.2 to the clean endpoint and denote the recovered latent by $\bar{v}_i^{(m)}$. We average over $M = 5$ fixed Gaussian noise realizations per part and define

$$Q_{\mathrm{prior}}(c) = \frac{1}{|\mathcal{O}_c|} \sum_{o \in \mathcal{O}_c} \frac{1}{|\mathcal{P}_o|} \sum_{i \in \mathcal{P}_o} \frac{1}{M} \sum_{m=1}^{M} \mathrm{CD}\left(G_\psi(\bar{v}_i^{(m)}), P_i\right). \tag{44}$$

The same object-level text condition c_o from Eq. 16 is supplied to the frozen prior. Because this diagnostic starts from encoded ground-truth geometry rather than the coarse latent predicted by the joint generator, it measures category-dependent prior recovery without introducing structural or autoregressive prediction errors.

Across the 12 held-out categories, Q_{AE} ranges from 0.31 to 0.61, with a median of 0.44, whereas Q_{prior} ranges from 0.92 to 1.68, with a median of 1.23. The frozen geometry modules therefore exhibit measurable variation in reconstruction and recovery quality across categories.

Category-wise ARTUS improvement. We next test whether categories with higher-quality frozen geometry modules systematically receive larger improvements from ARTUS. We consider two complementary measures.

First, we measure the improvement over ArtFormer, the strongest baseline in Table 1:

$$\Delta_{\mathrm{base}}(c) = \mathrm{CD}_{\mathrm{ArtFormer}}(c) - \mathrm{CD}_{\mathrm{ARTUS}}(c). \tag{45}$$

Larger positive values indicate larger improvements of ARTUS over the external baseline. This quantity reflects the complete difference between the two generation systems, including representation, fusion, autoregressive reasoning, and geometry refinement.

Second, we measure the improvement over the coarse-only ARTUS control:

$$\Delta_{\mathrm{refine}}(c) = \mathrm{CD}_{\mathrm{Coarse}}(c) - \mathrm{CD}_{\mathrm{ARTUS}}(c). \tag{46}$$

The coarse-only control follows Table 3 but is evaluated separately on each held-out category. Because the joint generator is unchanged, Δ_{refine} more directly characterizes the additional improvement associated with geometry refinement and pathway feedback.

Table 10 reports both geometry-quality measures and both ARTUS improvement measures for all 12 categories.

Correlation with ARTUS improvement. Category-wise geometry quality and model improvement need not follow a linear relationship. We therefore use Spearman rank correlation across the 12 semantic categories and evaluate four pairwise associations:

$$\begin{aligned} \rho_{\mathrm{AE,base}} &= \rho_S\left(Q_{\mathrm{AE}}, \Delta_{\mathrm{base}}\right), & \rho_{\mathrm{prior,base}} &= \rho_S\left(Q_{\mathrm{prior}}, \Delta_{\mathrm{base}}\right), \\ \rho_{\mathrm{AE,refine}} &= \rho_S\left(Q_{\mathrm{AE}}, \Delta_{\mathrm{refine}}\right), & \rho_{\mathrm{prior,refine}} &= \rho_S\left(Q_{\mathrm{prior}}, \Delta_{\mathrm{refine}}\right). \end{aligned} \tag{47}$$

Table 10: **Category-wise frozen-geometry quality and ARTUS improvement.** Q_{AE} is the reconstruction CD of the frozen geometry representation, and Q_{prior} is the CD after controlled partial-noise recovery by the frozen prior at $\zeta^* = 250$. Lower is better for both. Δ_{base} denotes the CD improvement of ARTUS over ArtFormer, while Δ_{refine} denotes the improvement over the ARTUS coarse-only control; higher is better for both. Values are averaged first over parts within each object and then over held-out objects within each category. CD-related quantities use the same 10^{-2} convention as Table 1.

Category	$Q_{\mathrm{AE}} \downarrow$	$Q_{\mathrm{prior}} \downarrow$	$\Delta_{\mathrm{base}} \uparrow$	$\Delta_{\mathrm{refine}} \uparrow$
Box	0.34	1.05	0.26	0.48
Laptop	0.31	0.92	0.08	0.22
Oven	0.42	1.18	0.12	0.25
Pliers	0.52	1.42	0.06	0.18
Refrigerator	0.45	1.35	0.30	0.55
Scissors	0.58	1.50	0.09	0.20
Stapler	0.38	1.08	0.22	0.30
StorageFurniture	0.47	1.28	0.35	0.72
Table	0.36	0.98	0.10	0.28
TrashCan	0.40	1.12	0.16	0.35
USB	0.61	1.68	0.07	0.15
WashingMachine	0.49	1.45	0.62	0.81

Table 11: **Correlation between frozen geometry quality and ARTUS improvement.** Spearman ρ is computed across the 12 held-out semantic categories. Confidence intervals are obtained by category-level bootstrap and p-values by a two-sided permutation test.

Variables	Spearman ρ	95% CI	p
Q_{AE} vs. Δ_{base}	-0.17	$-0.68, 0.69$	0.61
Q_{prior} vs. Δ_{base}	-0.07	$-0.65, 0.75$	0.83
Q_{AE} vs. Δ_{refine}	-0.27	$-0.79, 0.54$	0.40
Q_{prior} vs. Δ_{refine}	-0.18	$-0.76, 0.63$	0.58

Since Q_{AE} and Q_{prior} are error measures, a systematic tendency for better frozen geometry quality to yield larger ARTUS gains would appear as a negative correlation. We report 95% bootstrap confidence intervals using 10,000 category-level resamples and two-sided permutation-test p-values using 10,000 random permutations.

The category-wise correlations are weak across all four comparisons. The largest absolute rank correlation is 0.27, and the corresponding confidence intervals include zero. In particular, prior recovery quality has only a weak association with the improvement over ArtFormer ($\rho = -0.07$, 95% CI $[-0.65, 0.75]$, $p = 0.83$) and with the improvement over the coarse-only ARTUS control ($\rho = -0.18$, 95% CI $[-0.76, 0.63]$, $p = 0.58$). Autoencoder reconstruction quality exhibits the same pattern, with $\rho = -0.17$ for Δ_{base} and $\rho = -0.27$ for Δ_{refine}.

Frozen-prior quality can still affect absolute surface fidelity: categories with larger Q_{prior} may remain more difficult in terms of final CD. The correlation analysis addresses a different question—whether category-to-category variation in frozen-prior quality explains the *relative* improvement of ARTUS. Across the evaluated held-out categories, we observe no strong monotonic relationship between these quantities.

D.4 Sensitivity to loss weights

The joint objective in Eq. 11 combines four terms through the weights (α, β, λ). The main experiments use the validation-selected setting $(\alpha, \beta, \lambda) = (1, 0.5, 0.1)$ (Section 4). This section varies one weight at a time around that setting while holding the remaining two fixed: $\alpha \in \{0.5, 1, 2\}$, $\beta \in \{0.1, 0.5, 1.0\}$, and $\lambda \in \{0, 0.05, 0.1, 0.2\}$. Each configuration shares the frozen geometry modules, training data, optimization budget, and inference configuration with the full model, and is evaluated on complex multi-part objects with means across five training runs, following the protocol of Table 2.

Table 12 collects the sensitivity results. The validation-selected setting $(1, 0.5, 0.1)$ provides the best overall compromise across metrics. It is not optimal for every metric individually—for instance, $\alpha = 2$ attains a slightly lower layout error (d_{gIoU} 0.17 versus 0.18) at the cost of surface fidelity (CD 2.71). Lowering the structural weight ($\alpha = 0.5$) degrades layout and kinematic accuracy (d_{gIoU} 0.23, POR 0.14). Reducing the geometry weight to $\beta = 0.1$ increases CD to 2.89. Removing the structure-controlled latent loss ($\lambda = 0$) increases CD from 2.62 to 2.80 and POR from 0.10 to 0.13. These output-level differences support the auxiliary objective under the evaluated configuration.

Table 12: **Sensitivity to the loss weights** (α, β, λ)**.** One weight varies per row. The remaining two stay at the validation-selected setting $(1, 0.5, 0.1)$, highlighted in **bold**. All settings evaluate complex multi-part objects with identical frozen geometry modules and training budgets.

Varied	(α, β, λ)	CD ↓	F-score ↑	d_{gIoU} ↓	d_{cDist} ↓	CLIP ↑	POR ↓
α	$(0.5, 0.5, 0.1)$	2.78	73.20	0.23	0.19	0.30	0.14
α	$(\mathbf{1}, \mathbf{0.5}, \mathbf{0.1})$	**2.62**	**75.61**	**0.18**	**0.15**	**0.31**	**0.10**
α	$(2, 0.5, 0.1)$	2.71	74.45	0.17	0.15	0.29	0.11
β	$(1, 0.1, 0.1)$	2.89	71.80	0.21	0.18	0.27	0.13
β	$(1, 1.0, 0.1)$	2.65	75.10	0.20	0.17	0.31	0.12
λ	$(1, 0.5, 0)$	2.80	73.05	0.22	0.18	0.29	0.13
λ	$(1, 0.5, 0.05)$	2.68	74.80	0.19	0.16	0.30	0.11
λ	$(1, 0.5, 0.2)$	2.73	74.15	0.19	0.16	0.29	0.12

D.5 Stop-gradient control

The structure-controlled latent objective in Eq. 10 matches the predicted fused representation $\hat{r}_k$ to the representation r_k constructed from the ground-truth structural and geometric factors. The main model applies stop-gradient to r_k, so that the auxiliary objective is

$$\mathcal{L}_{\text{scl}}^{\text{sg}} = \frac{1}{|\mathcal{J}|} \sum_{(i,k)\in\mathcal{J}} \|\hat{r}_k - \text{sg}(r_k)\|_2^2 . \tag{48}$$

To examine whether this asymmetric gradient treatment is necessary, we replace Eq. 48 with a symmetric alternative,

$$\mathcal{L}_{\text{scl}}^{\text{sym}} = \frac{1}{|\mathcal{J}|} \sum_{(i,k)\in\mathcal{J}} \|\hat{r}_k - r_k\|_2^2 , \tag{49}$$

which has the same forward matching value for fixed model parameters but allows gradients from the auxiliary loss to propagate through both branches. All other model components, loss weights, training data, optimization budget, frozen geometry modules, and inference settings are kept unchanged.

Table 13 compares the three configurations on the complex multi-part subset over five training runs.

Table 13: **Stop-gradient control.** Comparison between removing the structure-controlled latent objective, symmetric representation matching, and the proposed stop-gradient target. All variants use identical architecture, loss weights apart from the stated change, training budget, frozen geometry modules, and inference settings. Values are mean ± standard deviation over five runs on complex multi-part objects.

Variant	CD ↓	F-score ↑	d_{gIoU} ↓	d_{cDist} ↓	CLIP ↑	POR ↓
w/o $\mathcal{L}_{\text{scl}}$	$2.80_{\pm 0.42}$	$73.05_{\pm 4.80}$	$0.22_{\pm 0.05}$	$0.18_{\pm 0.04}$	$0.29_{\pm 0.03}$	$0.13_{\pm 0.03}$
Symmetric	$2.70_{\pm 0.40}$	$74.30_{\pm 4.60}$	$0.20_{\pm 0.04}$	$0.16_{\pm 0.04}$	$0.30_{\pm 0.03}$	$0.12_{\pm 0.03}$
Stop-gradient (ARTUS)	$\mathbf{2.62}_{\pm 0.38}$	$\mathbf{75.61}_{\pm 4.50}$	$\mathbf{0.18}_{\pm 0.04}$	$\mathbf{0.15}_{\pm 0.04}$	$\mathbf{0.31}_{\pm 0.03}$	$\mathbf{0.10}_{\pm 0.03}$

The symmetric objective improves over removing $\mathcal{L}_{\text{scl}}$, indicating that matching the fused representation is beneficial even when both branches receive gradients. However, the stop-gradient formulation performs consistently better. Compared with symmetric matching, it reduces CD from 2.70 to 2.62, d_{gIoU} from 0.20 to 0.18, d_{cDist} from 0.16 to 0.15, and POR from 0.12 to 0.10, while increasing F-score from 74.30 to 75.61 and CLIP from 0.30 to 0.31.

With symmetric matching, both branches are updated by the auxiliary loss, whereas stop-gradient keeps r_k fixed for this update. The stronger results indicate that this anchored target provides a more effective training signal.

D.6 Gate activation statistics

The structure-derived gate g_i in Eq. 4 controls how much geometric evidence enters each joint token. Beyond the output-quality controls in Table 2, the gate values themselves can be inspected on the trained checkpoint. For every generated node we record the channel-wise gate $g_i \in (0,1)^d$ and summarize it by its channel mean and by the fraction of channels exceeding 0.5. Nodes are grouped by their root-to-node path depth, so that the statistics expose whether structural control behaves differently for shallow structural parts and deep moving parts. The statistics are computed on the test split with the validation-selected weights and describe the trained checkpoint.

Table 14 reports these statistics per depth group. Gate activation decreases with path depth: root-adjacent structural parts exhibit higher average gate values (0.54) and a majority of active channels (56.8%), whereas deeper nodes along the articulation tree exhibit progressively lower gate means (0.38 for Depth ≥ 3) and fewer active channels (31.5%). This trend is consistent with stricter geometric filtering for distal moving parts.

Table 14: **Gate activation statistics by hierarchy depth.** Channel-mean gate value, standard deviation, and the fraction of gate channels above 0.5, averaged over nodes within each root-to-node depth group across the test split.

Depth group	Mean gate	Std. gate	Channels > 0.5 (%)
Depth 1 (root-adjacent)	0.54	0.21	56.8
Depth 2	0.45	0.19	42.4
Depth ≥ 3	0.38	0.17	31.5
All nodes	0.46	0.19	44.1

D.7 Text-conditioning ablation

All main experiments condition generation on rendering-derived captions (Section C.1). To isolate the contribution of the text condition, we consider two controls under the evaluation protocol of Table 9: removing the text condition, and replacing each caption with a caption sampled from a different object (shuffled text). The full-text row reproduces the full-model values and is highlighted in **bold**.

Table 15 shows that removing the text prompt drops CLIP from 0.31 to 0.19 and increases CD to 2.88. Shuffled text degrades CD to 3.15 and POR to 0.16. These comparisons indicate that text guidance provides semantic constraints for both geometry and part arrangement.

Table 15: **Text-conditioning ablation.** All settings evaluate complex multi-part objects under the protocol of Table 9. The full-text row is highlighted in **bold**.

Condition	CD ↓	F-score ↑	d_{gIoU} ↓	d_{cDist} ↓	CLIP ↑	POR ↓
Full text	**2.62**	**75.61**	**0.18**	**0.15**	**0.31**	**0.10**
No text	2.88	72.15	0.21	0.18	0.19	0.13
Shuffled text	3.15	68.20	0.26	0.22	0.15	0.16

D.8 Aligned-Geometry Articulation Diagnostics

While main comparisons reflect both geometry and structure, we use an aligned-geometry protocol to inspect articulation predictions directly. Each method's part geometry is replaced by the same reference meshes in a common frame, while retaining its predicted part arrangements and motion. Because geometry is fully controlled under shared poses, visible differences isolate structural quality from surface appearance.

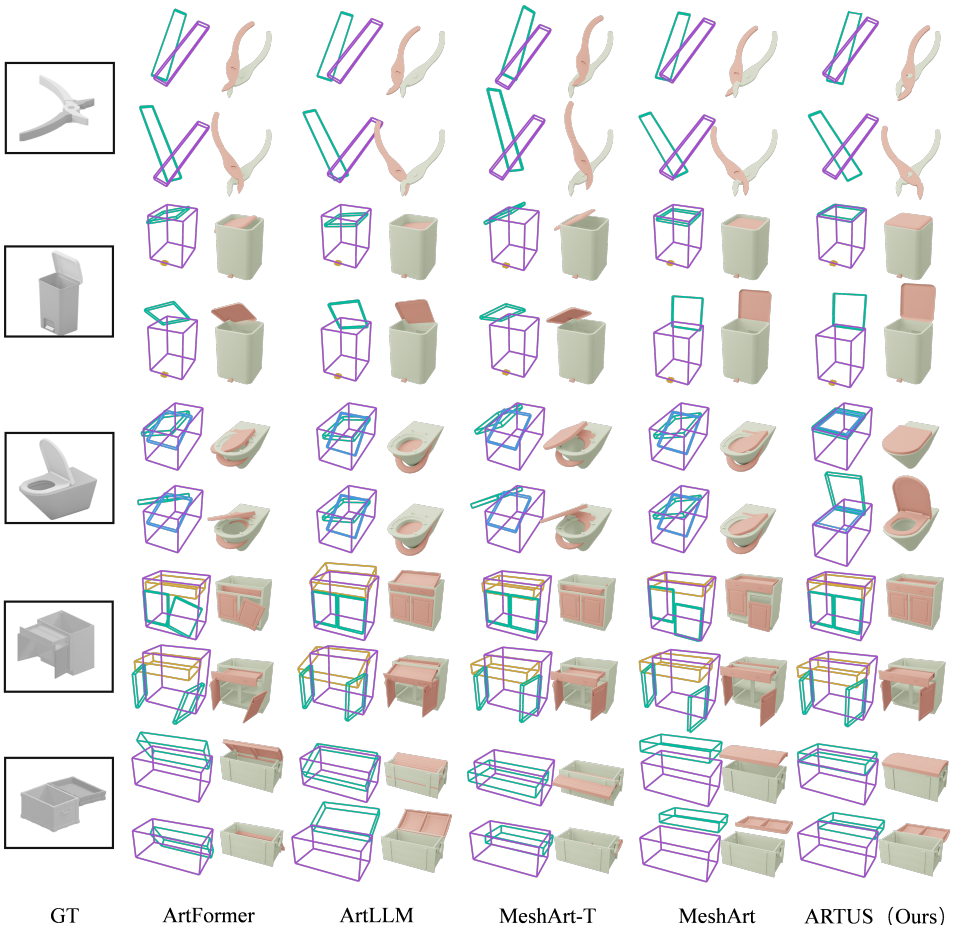

Figure A8: **Aligned-geometry articulation comparison.** Shared reference geometry removes appearance variation across methods, exposing differences in part layout, articulation, and inter-part overlap.

Figure A8 compares matched qualitative outputs. Wireframe boxes show part layouts, while posed geometry evaluates compatibility after motion. Baselines often appear plausible at rest but exhibit displacement or increased overlap once articulated; meanwhile, ARTUS maintains valid part arrangements across states, consistent with the lower layout errors and POR in Tables 1 and 2.

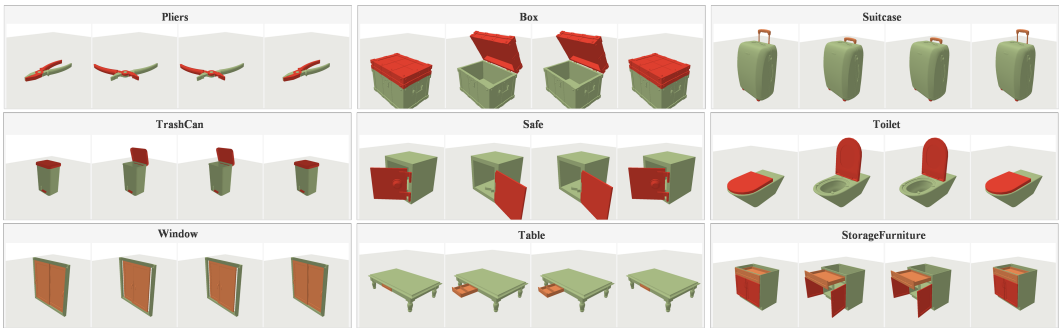

Figure A9: **ARTUS outputs across shared motion states.** Representative articulated objects generated by ARTUS, shown at four shared normalized motion states from rest to terminal articulation. Red parts are the moving components.

Figure A9 shows generated objects across the shared normalized motion fractions used by the motion protocol.