TAMING TRELLIS.2: TOPOLOGY-AWARE ONLINE POST-TRAINING FOR 3D MESH GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Directly generating 3D assets is essential for scalable 3D content creation. Recent latent flow-based 3D generative models, such as TRELLIS.2, achieve impressive generation quality and efficiency by employing expressive 3D representations that support arbitrary topology. However, such topological flexibility comes at the cost of losing guarantees on topological regularity, potentially introducing unintended non-manifold structures when regular topology is desired. We propose *TopoFT*, a topology-aware fine-tuning method for online post-training of pretrained 3D generators that improves topological regularity without modifying the underlying O-Voxel representation or restricting its support for arbitrary topology. Our diagnostic analysis identifies Stage-2 shape-SLat generation as a major source of topology irregularity. Since topology is evaluated globally while individual irregularities are spatially localized, TopoFT decomposes each mesh-level group-relative advantage into additive token-wise attributions that exactly preserve the global advantage. An anchored dual-branch flow objective then reinforces favorable local trajectories while stably suppressing unfavorable ones. Across four unseen benchmarks, TopoFT consistently reduces non-manifold edges, non-manifold vertices, and boundary edges without systematic degradation in geometric fidelity. These improvements translate into substantial downstream benefits: the target-reaching rate under aggressive QEM simplification increases from 52.7–91.0% to 95.8–99.4%, while the open-wall path ratio in downstream slicing decreases from 15.7–22.9% to 8.0–14.5%.

1 INTRODUCTION

Recent latent flow-based 3D models have achieved remarkable progress in high-quality image-to-3D generation. Rather than directly predicting explicit meshes, these methods generate expressive latent geometric representations that are subsequently decoded into meshes (Xiang et al., 2025; Tencent Hunyuan3D Team, 2025). While such representations provide strong geometric flexibility, the resulting meshes may contain topology irregularities, including non-manifold edges, non-manifold vertices, and boundary edges (Botsch et al., 2010). As shown in Fig. 1, these irregularities can be visually subtle but substantially affect downstream applications that require regular, manifold, or watertight geometry (Zheng et al., 2024; Wei et al., 2025).

A common solution is to repair generated meshes through post-processing (Wen et al., 2025; Huang et al., 2020). However, post-processing corrects individual outputs without improving the underlying generator. Topology-aware fine-tuning offers a more fundamental alternative, but is challenging for latent flow-based generators: *topology is evaluated only after mesh decoding, whereas optimization is performed over latent representations*. Moreover, *mesh-level topology feedback is global, while individual irregularities are spatially localized*. Uniformly assigning the same feedback to all latent tokens therefore provides an imprecise optimization signal.

In this work, we propose *TopoFT*, a topology-aware fine-tuning method for online post-training of TRELLIS.2. A controlled diagnostic study identifies Stage-2 shape-SLat generation as a major source of topology irregularity, motivating us to directly fine-tune the Stage-2 geometry flow model while leaving the underlying representation and decoding pipeline unchanged. To bridge the mismatch between global topology evaluation and local latent optimization, TopoFT decomposes the mesh-level group-relative advantage into spatial token-wise attributions that preserve the global ob-

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.

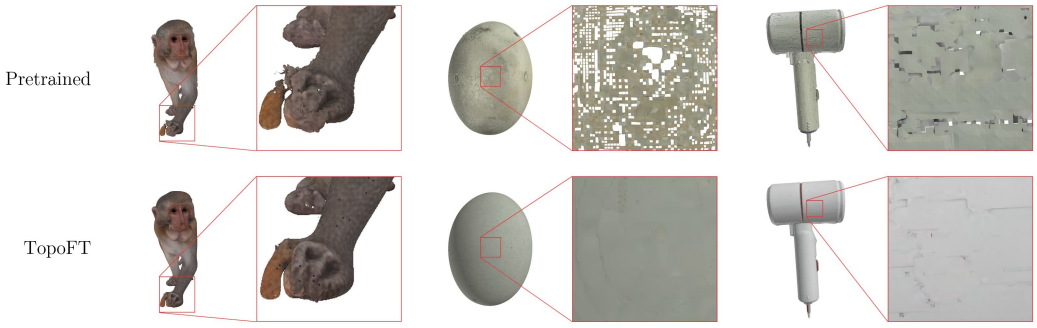

Figure 1: Qualitative comparison between pretrained TRELLIS.2 and TopoFT. Red boxes mark local topology irregularities, and zoomed-in views show their reduction with TopoFT. **See Supplementary Material for a rotating comparison video.**

jective. Directly applying negative token attributions to a squared flow-matching loss is ill-posed, as the objective becomes unbounded below. We therefore introduce an anchored dual-branch objective that reinforces favorable tokens toward their flow-matching targets while assigning unfavorable tokens finite anchor-relative targets for stable optimization.

Across four unseen benchmarks, TopoFT consistently improves topological regularity without systematic degradation in geometric fidelity. More importantly, these improvements translate into substantially better downstream processability, enabling more reliable mesh simplification and markedly reducing open-wall artifacts in slicing.

Our main contributions are summarized as follows:

- We systematically diagnose topology irregularities in the TRELLIS.2 generation pipeline and identify Stage-2 shape-SLat generation as a major source.
- We propose TopoFT, which transfers global mesh-level topology feedback to spatial shape-SLat tokens and introduces an anchored dual-branch flow objective for stable signed topology-aware optimization.
- Extensive experiments show substantial improvements in mesh topology while preserving geometric fidelity, together with clear gains in downstream mesh simplification and slicing.

2 Preliminaries and Diagnostic Analysis

2.1 TRELLIS.2 and Mesh Topology

TRELLIS.2 geometry generation. TRELLIS.2 generates geometry through three stages: Stage-1 predicts a sparse spatial structure, Stage-2 generates shape-SLat features over its active locations using a conditional flow model, and Stage-3 uses the pretrained Sparse Compression VAE (SC-VAE) decoder to map shape-SLat features to O-Voxel, from which the final mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ is extracted, where $\mathcal{V}$, $\mathcal{E}$, and $\mathcal{F}$ denote the sets of vertices, edges, and faces. Given an image condition c, let $\mathcal{S} = \{\mathbf{s}_j\}_{j=1}^{N}$ denote the active spatial locations produced by Stage-1 and $x_0 = \{x_{0,j}\}_{j=1}^{N}$ the corresponding shape-SLat tokens generated by Stage-2. Here, N denotes the number of active shape-SLat tokens. Stage-3 then maps x_0 to the final mesh through O-Voxel decoding and mesh extraction. The Stage-2 flow model, parameterized by θ, predicts token-wise velocities

$$v_\theta(x_t, t, c, \mathcal{S}) = \{v_{\theta,j}(x_t, t, c, \mathcal{S})\}_{j=1}^{N}, \tag{1}$$

where $t \in [0, 1]$ denotes the flow time, x_t denotes the noisy shape-SLat at time t, and $v_{\theta,j}$ denotes the predicted velocity of token j. These velocities are integrated from Gaussian noise to obtain x_0.

Mesh topology statistics. Following standard polygon-mesh definitions (Botsch et al., 2010), we consider non-manifold edges (NME), non-manifold vertices (NMV), and boundary edges (BND). An NME is incident to more than two faces, an NMV has incident faces that do not form a single

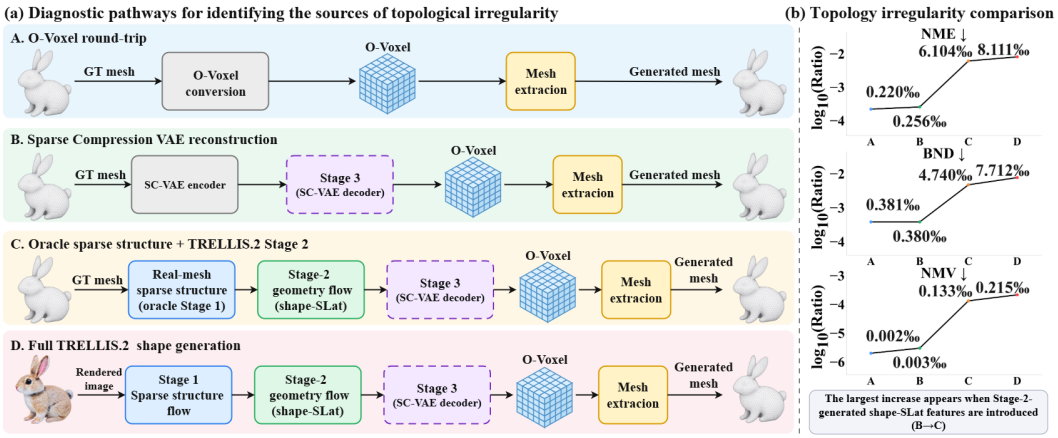

Figure 2: **Diagnosing topology irregularity in TRELLIS.2.** (a) Four controlled pipelines progressively isolate the contributions of O-Voxel meshing, SC-VAE reconstruction, Stage-2 shape-SLat generation, and full TRELLIS.2 generation on 5,000 topology-regular meshes. (b) NME, NMV, and BND ratios on logarithmic scales; annotations report the original values in per mille (‰). Introducing Stage-2-generated shape-SLat features causes the largest increase in topology irregularity, while Stage-1 introduces additional irregularity.

continuous local surface, and a BND is incident to exactly one face. Denoting their corresponding sets by $\mathcal{E}_{\text{NME}}$, $\mathcal{V}_{\text{NMV}}$, and $\mathcal{E}_{\text{BND}}$, we quantify the prevalence of these topology irregularities within each mesh using the normalized ratios

$$\rho_{\text{NME}} = |\mathcal{E}_{\text{NME}}| \,/\, |\mathcal{E}|, \qquad \rho_{\text{NMV}} = |\mathcal{V}_{\text{NMV}}| \,/\, |\mathcal{V}|, \qquad \rho_{\text{BND}} = |\mathcal{E}_{\text{BND}}| \,/\, |\mathcal{E}|. \tag{2}$$

Here, $|\cdot|$ denotes the cardinality of a set. Degenerate meshes with $|\mathcal{V}| = 0$ or $|\mathcal{E}| = 0$ are handled separately (App. B.2). These configurations are not inherently invalid: open or non-manifold structures may be necessary for representing arbitrary topology. We treat them as topology irregularities only when regular surface topology is desired.

2.2 Diagnosing the Sources of Topology Irregularity

Topology irregularity may be introduced at different stages of the TRELLIS.2 generation pipeline. To localize its primary source, we conduct a controlled study on 5,000 topology-regular meshes from the Fusion 360 Gallery Segmentation Dataset (Lambourne et al., 2021). As illustrated in Fig. 2, we progressively introduce O-Voxel meshing, SC-VAE reconstruction, Stage-2 generation, and full TRELLIS.2 generation while measuring the resulting topology irregularity. Further implementation details of the diagnostic pipelines are provided in App. B.1.

The dominant increase occurs when Stage-2-generated shape-SLat features are introduced: from pipeline B to C, the NME, NMV, and BND ratios increase by approximately $23.8\times$, $44.3\times$, and $12.5\times$. Stage-1 introduces an additional increase, whereas O-Voxel round-trip meshing and SC-VAE reconstruction contribute comparatively little. These results identify Stage-2 shape-SLat generation as a major source of topology irregularity and motivate us to post-train the Stage-2 geometry flow model while keeping Stage-1 and Stage-3 fixed. The spatial organization of shape-SLat tokens further provides a natural interface for localizing mesh-level topology feedback.

3 Method

3.1 Topology-Aware Online Post-Training

Topology irregularity cost. Let $\text{Dec}(\cdot)$ denote the fixed pipeline comprising Stage-3 decoding to O-Voxel and subsequent mesh extraction. Using the mesh topology statistics defined in Sec. 2, we measure the topology irregularity of a decoded mesh $\mathcal{M} = \text{Dec}(x_0)$ as

$$D_{\text{topo}}(\mathcal{M}) = w_1 \rho_{\text{NME}}(\mathcal{M}) + w_2 \rho_{\text{NMV}}(\mathcal{M}) + w_3 \rho_{\text{BND}}(\mathcal{M}), \tag{3}$$

where w_1, w_2, and w_3 control the relative importance of the three topology statistics. A lower D_{topo} indicates a more topologically regular mesh under the target regularity setting.

Complexity-preserving regularization. Optimizing normalized irregularity ratios alone admits degenerate solutions: the model may improve the ratios by substantially changing mesh complexity rather than correcting irregular connectivity. Before post-training, for each condition c we cache its Stage-1 sparse structure $\mathcal{S}$ and a non-empty reference mesh $\mathcal{M}^{\text{ref}} = (\mathcal{V}^{\text{ref}}, \mathcal{E}^{\text{ref}}, \mathcal{F}^{\text{ref}})$ generated by the frozen pretrained Stage-2 model under $\mathcal{S}$. Since vertex, edge, and face counts are strongly correlated, we use the edge count as a proxy for mesh complexity and penalize its relative deviation from the reference:

$$D_{\text{comp}}\left(\mathcal{M}, \mathcal{M}^{\text{ref}}\right) = w_4 \frac{\left||\mathcal{E}^{\text{ref}}| - |\mathcal{E}|\right|}{|\mathcal{E}^{\text{ref}}|}. \tag{4}$$

where w_4 controls the regularization strength; removing this term causes complexity collapse (App. A.3). The resulting regularized topology cost is

$$D(x_0) = D_{\text{topo}}(\mathcal{M}) + D_{\text{comp}}(\mathcal{M}, \mathcal{M}^{\text{ref}}). \tag{5}$$

Topology-guided online objective. Let $\mathcal{C}$ denote the dataset of conditioning images. Let $p_\theta(x_0 \mid c, \mathcal{S})$ denote the Stage-2 shape-SLat distribution conditioned on image c and the Stage-1 sparse structure $\mathcal{S}$. We seek to minimize the expected regularized topology cost:

$$\min_\theta \ \mathbb{E}_{c\sim\mathcal{C}}\mathbb{E}_{x_0\sim p_\theta(\cdot|c,\mathcal{S})}\left[D(x_0)\right]. \tag{6}$$

Since D is non-differentiable through mesh extraction, we optimize a group-relative flow-matching surrogate.

At the beginning of each online round, we freeze the current model as θ_{old} and use it for sampling. For each condition c, we obtain a shared sparse structure $\mathcal{S}$ and generate K shape-SLat samples $\{x_0^{(i)}\}_{i=1}^K \sim p_{\theta_{\text{old}}}(\cdot \mid c, \mathcal{S})$. After decoding them into meshes, we compute their regularized topology costs $D^{(i)} = D(x_0^{(i)})$ and define the group-relative advantage

$$A^{(i)} = \frac{\bar{D} - D^{(i)}}{\sigma_D + \zeta}, \tag{7}$$

where $\bar{D}$ and σ_D are the mean and standard deviation of the K group costs, and $\zeta > 0$ is a small constant for numerical stability. Thus, $A^{(i)} > 0$ favors samples with lower-than-average topology cost, whereas $A^{(i)} < 0$ suppresses samples with higher-than-average cost.

3.2 Mesh-to-Token Topology Attribution

The group-relative advantage $A^{(i)}$ provides a global evaluation of the i-th generated mesh. However, topology irregularities are spatially localized, whereas the Stage-2 flow-matching objective is defined over individual shape-SLat tokens. We therefore decompose the mesh-level cost, and consequently its group-relative advantage, into spatially aligned token-wise contributions.

Mesh-to-token cost decomposition. Each mesh vertex is associated with an active O-Voxel location. Given the shape-SLat and O-Voxel grid resolutions R and G, we map an O-Voxel coordinate $\mathbf{p}$ to the shape-SLat grid by $\pi(\mathbf{p}) = \lfloor \frac{R}{G}\mathbf{p} \rfloor$. Through this mapping, each mesh vertex is assigned to one active shape-SLat location. For a vertex set $\mathcal{V}'$ and an edge set $\mathcal{E}'$, we decompose their counts to token j by

$$C_j(\mathcal{V}') = \sum_{u\in\mathcal{V}'} \mathbf{1}[\pi(\mathbf{p}_u) = \mathbf{s}_j], \quad C_j(\mathcal{E}') = \frac{1}{2}\sum_{(u,v)\in\mathcal{E}'} \left(\mathbf{1}[\pi(\mathbf{p}_u) = \mathbf{s}_j] + \mathbf{1}[\pi(\mathbf{p}_v) = \mathbf{s}_j]\right). \tag{8}$$

Here, $\mathbf{1}[\cdot]$ is the indicator function, equal to 1 when the enclosed condition holds and 0 otherwise. By construction, $\sum_j C_j(\mathcal{V}') = |\mathcal{V}'|$ and $\sum_j C_j(\mathcal{E}') = |\mathcal{E}'|$.

To decompose the complexity regularizer in Eq. 4, we define $s^{(i)} = \text{sign}(|\mathcal{E}^{\text{ref}}| - |\mathcal{E}^{(i)}|)$, with $\text{sign}(0) = 0$. The token-wise additive contribution is then

$$q_j^{(i)} = w_1 \frac{C_j(\mathcal{E}^{(i)}_{\text{NME}})}{|\mathcal{E}^{(i)}|} + w_2 \frac{C_j(\mathcal{V}^{(i)}_{\text{NMV}})}{|\mathcal{V}^{(i)}|} + w_3 \frac{C_j(\mathcal{E}^{(i)}_{\text{BND}})}{|\mathcal{E}^{(i)}|} + w_4 s^{(i)} \frac{C_j(\mathcal{E}^{\text{ref}}) - C_j(\mathcal{E}^{(i)})}{|\mathcal{E}^{\text{ref}}|}. \tag{9}$$

Importantly, $q_j^{(i)}$ is not an independently normalized local topology cost and need not be non-negative. Instead, it is an additive contribution to the same global regularized cost. Using Eq. 8, the decomposition exactly preserves the sample-level objective, i.e., $D^{(i)} = \sum_{j=1}^{N} q_j^{(i)}$.

Global-advantage-preserving token attribution. For the K samples generated under the same condition and sparse structure, let $\bar{q}_j = \frac{1}{K}\sum_{i=1}^{K} q_j^{(i)}$ denote the group-average contribution at token j. Since $D^{(i)} = \sum_j q_j^{(i)}$, we have $\bar{D} = \sum_j \bar{q}_j$. The sample-level group-relative advantage therefore admits the exact decomposition

$$A^{(i)} = \frac{\bar{D} - D^{(i)}}{\sigma_D + \zeta} = \sum_{j=1}^{N} a_j^{(i)}, \qquad a_j^{(i)} = \frac{\bar{q}_j - q_j^{(i)}}{\sigma_D + \zeta}. \tag{10}$$

All token attributions share the global cost scale σ_D rather than using token-specific normalization. This preserves the scale of token-level cost contributions, allowing larger cost differences to produce correspondingly larger attributions. A token-specific normalization could instead assign large values to regions with little global contribution simply because their local variance is small.

Accordingly, $a_j^{(i)}$ is not an independently estimated local advantage, but an additive token-level attribution of the global advantage $A^{(i)}$. Its sign indicates whether token j contributes favorably or unfavorably to the global topology objective relative to its group-average contribution, while $|a_j^{(i)}|$ quantifies the magnitude of its additive contribution to $A^{(i)}$.

3.3 Anchored Dual-Branch Flow Optimization

Given the token-wise advantage attributions $a_j^{(i)}$ from Sec. 3.2, we use their signs and magnitudes to determine the direction and strength of flow-matching supervision for individual shape-SLat tokens. We omit explicit dependence on c and $\mathcal{S}$ when clear from context.

For the i-th sampled shape-SLat $x_0^{(i)}$, we sample $t^{(i)} \sim p(t)$ from a `logit-normal` distribution (see App. B.2) and $\epsilon^{(i)} \sim \mathcal{N}(0, I)$ and construct

$$x_t^{(i)} = (1 - t^{(i)})x_0^{(i)} + t^{(i)}\epsilon^{(i)}, \qquad v^{\star,(i)} = \epsilon^{(i)} - x_0^{(i)}, \tag{11}$$

where $\epsilon^{(i)}$ is standard Gaussian noise, $t^{(i)}$ is the flow-matching timestep, $x_t^{(i)}$ is the corresponding noisy latent, and $v^{\star,(i)}$ denotes the target velocity. Let $v_{\theta,j}^{(i)}$ and $v_j^{\star,(i)}$ denote the corresponding Stage-2 prediction and target velocity of token j, and let

$$v_{\text{old},j}^{(i)} = \text{sg}\Big[v_{\theta_{\text{old}},j}^{(i)}(x_t^{(i)}, t^{(i)}, c, \mathcal{S})\Big] \tag{12}$$

denote the stop-gradient (sg) prediction of the frozen sampling model for the current online round.

Positive local matching. For $a_j^{(i)} \geq 0$, token j contributes favorably relative to the group mean. We therefore reinforce the corresponding sample by minimizing $a_j^{(i)}\|v_{\theta,j}^{(i)} - v_j^{\star,(i)}\|_2^2$. Thus, larger positive attribution produces stronger supervision toward the flow-matching target.

Anchored negative matching. For $a_j^{(i)} < 0$, directly optimizing the formal signed objective $a_j^{(i)}\|v_{\theta,j}^{(i)} - v_j^{\star,(i)}\|_2^2$ is ill posed, since it is unbounded below. We instead construct an anchor-relative negative target

$$v_{\text{neg},j}^{(i)} = v_{\text{old},j}^{(i)} + a_j^{(i)}\left(v_j^{\star,(i)} - v_{\text{old},j}^{(i)}\right), \qquad a_j^{(i)} < 0. \tag{13}$$

Since $a_j^{(i)} < 0$, this target lies on the ray extending from $v_{\text{old},j}^{(i)}$ away from $v_j^{\star,(i)}$, with $|a_j^{(i)}|$ controlling the magnitude. The negative branch minimizes $\|v_{\theta,j}^{(i)} - v_{\text{neg},j}^{(i)}\|_2^2$, which is lower bounded and admits the finite minimizer $v_{\text{neg},j}^{(i)}$.

Unified signed local update. Combining both branches, we optimize

$$\ell_j^{(i)} = \begin{cases} a_j^{(i)}\left\|v_{\theta,j}^{(i)} - v_j^{\star,(i)}\right\|_2^2, & a_j^{(i)} \geq 0, \\ \left\|v_{\theta,j}^{(i)} - v_{\text{neg},j}^{(i)}\right\|_2^2, & a_j^{(i)} < 0. \end{cases} \tag{14}$$

Algorithm 1 Topology-Aware Online Post-Training (TopoFT)

Require: Pretrained Stage-2 model θ_{ref}, frozen Stage-1 and Stage-3, conditioning dataset $\mathcal{C}$, cached references $\{(\mathcal{S}, \mathcal{M}^{\text{ref}})\}$, group size K, learning rate η

$\theta \leftarrow \theta_{\text{ref}}$
for each online round **do**
  Freeze a copy of the current model: $\theta_{\text{old}} \leftarrow \theta$
  for each sampled condition $c \sim \mathcal{C}$ **do**
    Retrieve cached $(\mathcal{S}, \mathcal{M}^{\text{ref}})$
    Sample $x_0^{(i)} \sim p_{\theta_{\text{old}}}(\cdot \mid c, \mathcal{S})$, $i = 1, \dots, K$
    Decode $\{x_0^{(i)}\}_{i=1}^K$ and compute regularized topology costs $\{D^{(i)}\}_{i=1}^K$ using $\mathcal{M}^{\text{ref}}$
    Compute token-wise contributions $\{q_j^{(i)}\}$, advantage attributions $\{a_j^{(i)}\}$ using Sec. 3.2
    Store $\{c, \mathcal{S}, x_0^{(i)}, a_j^{(i)}\}_{i=1}^K$
  end for
  for each mini-batch **do**
    Construct flow-matching pairs $(x_t^{(i)}, v^{\star,(i)})$
    Compute the anchored dual-branch loss using Sec. 3.3
    Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{TopoFT}}$
  end for
end for
Output: Fine-tuned Stage-2 model θ

Proposition 1 (Global-to-local and signed-gradient preservation)**.** *TopoFT satisfies the following two properties.*

(i) Exact global-to-local preservation. The token-wise cost contributions and advantage attributions in Eqs. 9 and 10 exactly preserve their corresponding global quantities:

$$D^{(i)} = \sum_{j=1}^{N} q_j^{(i)}, \qquad A^{(i)} = \sum_{j=1}^{N} a_j^{(i)}. \tag{15}$$

(ii) Anchored signed-gradient preservation. For $a_j^{(i)} < 0$, the negative branch in Eq. 14 is lower bounded and admits the finite minimizer $v_{\text{neg},j}^{(i)}$. Moreover, for either branch, at the sampling-model anchor,

$$\nabla_{v_{\theta,j}^{(i)}} \ell_j^{(i)} \Big|_{v_{\theta,j}^{(i)} = v_{\text{old},j}^{(i)}} = 2a_j^{(i)} \left(v_{\text{old},j}^{(i)} - v_j^{\star,(i)} \right), \tag{16}$$

which is exactly the gradient of the formal signed objective $a_j^{(i)} \| v_{\theta,j}^{(i)} - v_j^{\star,(i)} \|_2^2$ at the same anchor.

Thus, positive attribution reinforces the sampled flow-matching trajectory, whereas negative attribution suppresses it through a finite anchor-relative target.

This equivalence is local to the sampling-model anchor. Away from θ_{old}, the negative branch approaches the finite target $v_{\text{neg},j}^{(i)}$ rather than performing unconstrained gradient reversal. Refreshing θ_{old} at each online round keeps the surrogate local to the evolving sampling distribution.

Overall objective. The TopoFT objective for a candidate group can be written as $\mathcal{L}_{\text{TopoFT}} = \frac{1}{KN} \sum_{i=1}^{K} \sum_{j=1}^{N} \ell_j^{(i)}$, where all K candidates share the same Stage-1 sparse structure and thus the same N active shape-SLat tokens. The complete online procedure is summarized in Algorithm 1.

4 Experiments

We organize our experiments around three questions. First, after being post-trained on one source dataset, **can TopoFT generalize to unseen object domains and consistently reduce topology irregularities**, measured by NME, NMV, and BND? Second, **does improving topology preserve the geometric fidelity of the generated surfaces**? Third, **do these topology improvements translate into meaningful gains in downstream mesh-processing tasks**? We answer these questions

through cross-dataset topology evaluation, geometric-fidelity analysis, and downstream experiments on mesh simplification and slicing.

4.1 Experimental Setup

Training. We fine-tune the Stage-2 geometry flow model of TRELLIS.2-4B (Xiang et al., 2026) using a training pool of 30,000 conditioning images rendered from 5,000 aesthetically filtered Objaverse-XL (Deitke et al., 2023) objects, with six views per object. An additional 200 disjoint objects are reserved for validation, with one randomly selected view per object. For both TopoFT variants, each conditioning image generates $K = 8$ shape-SLat samples from the current Stage-2 geometry flow model while sharing the same frozen Stage-1 sparse structure. All learned methods use the same data splits, optimization budget, and three training seeds. We use $(w_1, w_2, w_3, w_4) = (0.3, 0.2, 0.5, 0.1)$ throughout all experiments, with moderately greater emphasis on boundary edges due to their direct relevance to surface closure and downstream slicing. We keep these weights fixed across datasets; a sensitivity analysis is provided in App. A.7. Other training details are in App. B.2.

Baselines and model selection. We compare TopoFT with the pretrained TRELLIS.2-4B model and an offline DRO baseline adapted from DSO (Li et al., 2025). DSO optimizes samples generated offline by the pretrained Stage-2 geometry flow model, whereas TopoFT continually regenerates samples from the evolving Stage-2 geometry flow model. For all learned methods, checkpoints are selected on the same held-out validation set. Further details are in App. B.2 and App. B.3.

Test benchmarks and evaluation. We evaluate on four benchmarks spanning substantially different geometric domains: 1,000 real-scanned everyday objects from Google Scanned Objects (GSO) (Downs et al., 2022), 5,366 human-designed CAD models from the official Fusion 360 Gallery Segmentation test split (Lambourne et al., 2021), 5,911 category-diverse real-world objects from OmniObject3D (Wu et al., 2023), and 3,229 toy-shaped objects from Toys4K (Stojanov et al., 2021). The Fusion 360 test split is disjoint from the diagnostic subset used in Sec. 2. For each test object, all methods use the same conditioning image, Stage-1 sparse structure, and Stage-2 generation seed. Per-mesh metrics are computed independently for each evaluated mesh and then averaged within each benchmark. For compactness, Tables 2 and 3 report the mean across the three training seeds only. In the remaining result tables, learned methods are reported as mean$\pm$std over three training seeds, while pretrained TRELLIS.2-4B is evaluated once under the same paired protocol.

4.2 Generalization of Topology Regularization

Main topology results. Tab. 1 compares pretrained TRELLIS.2, DSO, and TopoFT. Relative to pretrained TRELLIS.2, token-level TopoFT achieves a $3.3\times$–$5.5\times$ reduction in NME, a $5.3\times$–$21.7\times$ reduction in NMV, and a $1.8\times$–$3.1\times$ reduction in BND. Token-level TopoFT also consistently outperforms the DSO baseline across all benchmarks and topology metrics. These results show that the topology improvements generalize across distinct geometric domains and different types of topology irregularities. Absolute irregularity counts are reported in App. A.1, and additional qualitative comparisons of mesh topology are in App. A.6.

Effect of mesh-to-token attribution. We compare TopoFT with a mesh-level variant that uses the same topology signal and anchored dual-branch optimization but uniformly assigns $A^{(i)}/N$ to all tokens instead of using token-wise attributions $a_j^{(i)}$. Relative to this variant, token-level attribution reduces NME by $1.3\times$–$1.8\times$, NMV by $2.5\times$–$4.0\times$, and BND by $1.2\times$–$1.5\times$. These results demonstrate the benefit of spatially localized topology feedback.

4.3 Geometric Fidelity

We next examine whether improving mesh topology comes at the cost of geometric fidelity. As shown in Tab. 2, TopoFT preserves the geometric fidelity of pretrained TRELLIS.2, with small improvements across all three metrics. These results indicate that the reduction in topology irregularity does not introduce an evident geometry–topology trade-off under our evaluation protocol. Complete results for all compared methods are in App. A.4, and detailed evaluation protocols are in App. B.4.1. Mesh-complexity statistics are reported in App. A.2 and confirm that the topology improvements are not obtained through substantial changes in mesh complexity.

Table 1: **Mesh generalization results.** We report normalized ratios (‰) of NME, NMV, and BND. Values in parentheses indicate the relative factor with respect to pretrained TRELLIS.2.

Dataset	Method	NME (‰) ↓	NMV (‰) ↓	BND (‰) ↓
GSO	TRELLIS.2	13.20	0.88	10.23
	DSO	9.30 ± 0.34 (1.4×)	0.48 ± 0.02 (1.8×)	9.39 ± 0.20 (1.1×)
	TopoFT(mesh)	5.65 ± 0.09 (2.3×)	0.33 ± 0.03 (2.7×)	6.84 ± 0.26 (1.5×)
	TopoFT(token)	$\mathbf{3.59 \pm 0.44}$ (3.7×)	$\mathbf{0.10 \pm 0.06}$ (8.8×)	$\mathbf{5.23 \pm 0.83}$ (2.0×)
Fusion 360	TRELLIS.2	7.46	0.21	4.99
	DSO	4.77 ± 0.22 (1.6×)	0.16 ± 0.01 (1.3×)	5.20 ± 0.23 (1.0×)
	TopoFT(mesh)	2.97 ± 0.02 (2.5×)	0.10 ± 0.01 (2.1×)	3.39 ± 0.07 (1.5×)
	TopoFT(token)	$\mathbf{2.23 \pm 0.27}$ (3.3×)	$\mathbf{0.04 \pm 0.02}$ (5.3×)	$\mathbf{2.75 \pm 0.18}$ (1.8×)
Omni Object3D	TRELLIS.2	20.96	1.30	14.89
	DSO	11.40 ± 0.43 (1.8×)	0.52 ± 0.02 (2.5×)	11.10 ± 0.40 (1.3×)
	TopoFT(mesh)	5.86 ± 0.13 (3.6×)	0.24 ± 0.00 (5.4×)	6.94 ± 0.22 (2.1×)
	TopoFT(token)	$\mathbf{3.81 \pm 0.55}$ (5.5×)	$\mathbf{0.06 \pm 0.04}$ (21.7×)	$\mathbf{4.77 \pm 0.97}$ (3.1×)
Toys4K	TRELLIS.2	20.65	1.26	12.29
	DSO	13.25 ± 0.44 (1.6×)	0.88 ± 0.04 (1.4×)	12.06 ± 0.22 (1.0×)
	TopoFT(mesh)	9.67 ± 0.17 (2.1×)	0.76 ± 0.03 (1.7×)	9.65 ± 0.21 (1.3×)
	TopoFT(token)	$\mathbf{5.44 \pm 1.12}$ (3.8×)	$\mathbf{0.23 \pm 0.15}$ (5.5×)	$\mathbf{6.77 \pm 1.36}$ (1.8×)

Table 2: **Geometric fidelity of pretrained TRELLIS.2 and TopoFT.** We report Chamfer distance, P95 distance, and F-score@0.01.

	Chamfer ($\times 10^{-2}$) ↓		P95 ($\times 10^{-2}$) ↓		F-score@0.01 (%) ↑	
Dataset	TRELLIS.2	**TopoFT**	TRELLIS.2	**TopoFT**	TRELLIS.2	**TopoFT**
GSO	6.02	**5.96**	31.82	**31.74**	6.34	**6.51**
Fusion 360	9.03	**8.98**	36.87	**36.76**	6.27	**6.48**
OmniObject3D	4.90	**4.87**	28.02	**28.00**	6.20	**6.26**
Toys4K	6.22	**6.19**	31.90	**31.84**	5.78	**5.91**

4.4 Downstream Mesh Processing

We next examine whether the topological improvements of TopoFT translate into practical benefits for downstream mesh processing. We consider two applications—quadric error metric (QEM) mesh simplification (Garland & Heckbert, 1997) and 3D printing slicing via CuraEngine (Ultimaker, 2016)—both applied directly to the generated meshes without any topology repair. For QEM, we report the target-reaching rate under a 10% face-count target and the achieved face ratio after simplification. For Cura slicing, we report the open-wall ratio (OWR) using a $0.1\,\mathrm{mm}$ endpoint-gap threshold and the affected-layer ratio (AL), defined as the fraction of slicing layers containing at least one open-wall artifact.

As shown in Tab. 3, TopoFT reaches the QEM simplification target more reliably, produces face ratios closer to the prescribed 10% target, and substantially reduces both open-wall artifacts and the fraction of affected slicing layers. These results demonstrate that improved mesh topology translates into better downstream processability. Complete results for all compared methods and additional downstream metrics are in App. A.5. Experimental settings, implementation details, and metric definitions are in App. B.4.2. Representative examples are shown in Fig. 3.

5 Related Work

3D Mesh Generation. Recent mesh generators represent geometry through discrete tokens, autoregressive sequences, or continuous latent representations (Weng et al., 2025; Chen et al., 2025; Lionar et al., 2025; Lin et al., 2026; Li et al., 2026). TRELLIS and TRELLIS.2 instead generate structured 3D latents that are subsequently decoded into meshes (Xiang et al., 2025; 2026). Recent methods

Table 3: **Downstream mesh processing.** Comparison between pretrained TRELLIS.2 and the final token-level TopoFT on QEM simplification and Cura slicing. For QEM, the target face ratio is 10%.

	QEM simplification				Cura slicing			
	Target reached (%) ↑		Face ratio (%) ↓		OWR (%) ↓		AL (%) ↓	
Dataset	TRELLIS.2	**TopoFT**	TRELLIS.2	**TopoFT**	TRELLIS.2	**TopoFT**	TRELLIS.2	**TopoFT**
GSO	73.90	**98.83**	12.23	**10.05**	22.92	**14.46**	49.35	**36.17**
Fusion 360	90.98	**99.44**	11.22	**10.05**	16.65	**9.43**	29.68	**19.94**
OmniObject3D	52.68	**99.33**	14.70	**10.04**	15.65	**7.98**	30.41	**18.96**
Toys4K	62.09	**95.75**	14.79	**10.25**	21.90	**11.83**	41.34	**29.12**

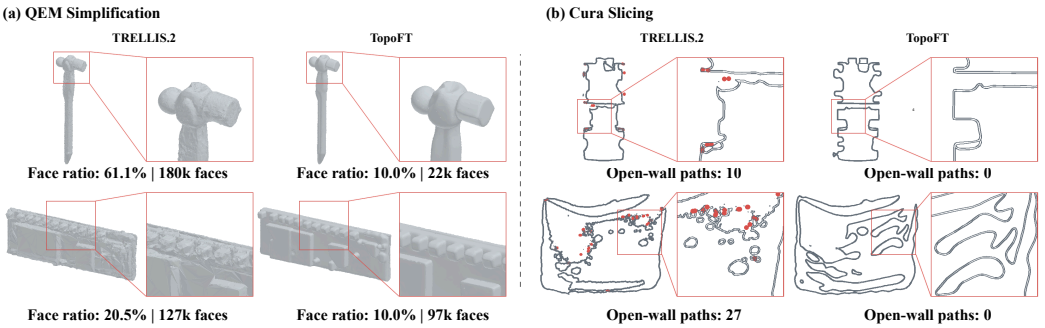

Figure 3: **Downstream mesh processing.** Compared with pretrained TRELLIS.2, TopoFT improves QEM simplification and reduces open-wall discontinuities during Cura slicing. "Open-wall paths" counts contour paths in the displayed slicing layer whose endpoint gap exceeds the 0.1 mm threshold used in quantitative evaluation. **Zoom in for details.**

have also incorporated topology-aware structures or connectivity constraints into mesh generation and reconstruction. LATO and LATO.2 model topology through structured latent representations and connectivity generation (Zhao et al., 2026; Long et al., 2026), while TopoMesh and ExMesh address topology during mesh autoencoding and reconstruction (Luo et al., 2026; Fan et al., 2026). A separate line of work repairs non-manifold geometry or reconstructs manifold surfaces from imperfect meshes (Hétroy et al., 2011; Huang et al., 2020). Unlike these approaches, TopoFT does not redesign the representation, decoder, or mesh extraction procedure, but instead improves an existing generator through topology-aware post-training.

Generative Post-Training. Reward- and preference-based post-training has been widely studied for diffusion and flow-based generative models (Black et al., 2024; Wallace et al., 2024; Wang et al., 2026), and recently extended to 3D generation; for example, DSO aligns pretrained 3D generators using object-level simulation feedback (Li et al., 2025). A key challenge is credit assignment: sample-level rewards do not reveal which parts of the representation contribute to the measured quality. Recent work explores finer-grained credit assignment during generation (He et al., 2026). TopoFT instead derives token-level credit directly from decoded mesh-topology statistics, without semantic annotations or an auxiliary learned reward model.

6 Conclusion

We presented TopoFT, an online post-training framework that improves the topological regularity of pretrained latent 3D generators without changing their representation or decoder. A controlled diagnosis identifies Stage-2 shape-SLat generation as a major source of topology irregularity, motivating token-level topology attribution and anchored dual-branch flow optimization. TopoFT substantially reduces topology irregularity across datasets while preserving geometric fidelity, and improves downstream mesh simplification and slicing. It does not enforce strict manifoldness, leaving hard topology guarantees for future work.

AI USE STATEMENT

OpenAI ChatGPT (GPT-5.6 Sol) was used as an assistive tool during this work. ChatGPT was used primarily for language editing and literature search assistance. References suggested during AI-assisted literature search were manually checked against the original sources before being included in the manuscript.

All final methodological choices, experimental protocols, evaluation criteria, and interpretations were determined by the authors. AI-assisted text, citations, and suggestions were reviewed and verified before inclusion. All reported experiments and numerical results were obtained by executing the authors' experimental pipelines.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon Mesh Processing*. A K Peters, Natick, Massachusetts, 2010. ISBN 978-1-56881-426-1.

Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. MeshAnything V2: Artist-created mesh generation with adjacent mesh tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13922–13931, 2025.

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

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560, 2022.

Chuanjin Fan, Lifan Wu, Wenjie Chang, Hanzhi Chang, Wenfei Yang, and Tianzhu Zhang. Exmesh: Explicit mesh reconstruction with topology adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27136–27145, 2026.

Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pp. 209–216, 1997.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 132869–132895, 2026.

Franck Hétroy, Stéphanie Rey, Carlos Andújar, Pere Brunet, and Àlvar Vinacua. Mesh repair with user-friendly topology control. *Computer-Aided Design*, 43(1):101–113, 2011.

Jingwei Huang, Yichao Zhou, and Leonidas J. Guibas. ManifoldPlus: A robust and scalable watertight manifold surface generation method for triangle soups. *arXiv preprint arXiv:2005.11621*, 2020.

Joseph G. Lambourne, Karl D. D. Willis, Pradeep Kumar Jayaraman, Aditya Sanghi, Peter Meltzer, and Hooman Shayani. Brepnet: A topological message passing system for solid models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12773–12782, 2021.

Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. DSO: Aligning 3d generators with simulation feedback for physical soundness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6772–6783, 2025.

Weiyu Li, Antoine Toisoul, Tom Monnier, Roman Shapovalov, Rakesh Ranjan, Ping Tan, and Andrea Vedaldi. MeshFlow: Efficient artistic mesh generation via MeshVAE and flow-based diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5849–5858, 2026.

Junkai Lin, Hang Long, Huipeng Guo, Jielei Zhang, Jiayi Yang, Tianle Guo, Yang Yang, Jianwen Li, Wenxiao ZHANG, Matthias Nießner, and Wei Yang. Meshripple: Structured autoregressive generation of artist-meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12706–12718, June 2026.

Stefan Lionar, Jiabin Liang, and Gim Hee Lee. TreeMeshGPT: Artistic mesh generation with autoregressive tree sequencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26608–26617, 2025.

Hang Long, Tianhao Zhao, Junkai Lin, Youjia Zhang, Huipeng Guo, Rendong Liang, Jiale Xu, Jozef Hladký, Matthias Nießner, Yuanming Hu, and Wei Yang. LATO.2: Factorized 3d mesh generation with vertex and topology flow. *arXiv preprint arXiv:2607.10623*, 2026.

Guan Luo, Xiu Li, Rui Chen, Xuanyu Yi, Jing Lin, Chia Hao Chen, Jiahang Liu, Song-Hai Zhang, and Jianfeng Zhang. TopoMesh: High-fidelity mesh autoencoding via topological unification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27082–27092, 2026.

Stefan Stojanov, Anh Thai, and James M. Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1798–1808, 2021.

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

Ultimaker. CuraEngine: A powerful, fast and robust engine for converting 3d models into g-code instructions. `https://github.com/Ultimaker/CuraEngine`, 2016. Accessed: 2026-09-25.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Yiyang Wang, Xi Chen, Xiaogang Xu, Yu Liu, and Hengshuang Zhao. GDRO: Group-level reward post-training suitable for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43505–43513, 2026.

Xinyue Wei, Fanbo Xiang, Sai Bi, Anpei Chen, Kalyan Sunkavalli, Zexiang Xu, and Hao Su. NeuManifold: Neural watertight manifold reconstruction with efficient and high-quality rendering support. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 731–741, 2025.

Huibiao Wen, Guilong He, Rui Xu, Shuangmin Chen, Shiqing Xin, Zhenyu Shu, Taku Komura, Jieqing Feng, Wenping Wang, and Changhe Tu. Feature-preserving mesh repair via restricted power diagram. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. Association for Computing Machinery, 2025.

Haohan Weng, Zibo Zhao, Biwen Lei, Xianghui Yang, Jian Liu, Zeqiang Lai, Zhuo Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, Tong Zhang, Shenghua Gao, and C. L. Philip Chen. Scaling mesh generation via compressive tokenization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11093–11103, 2025.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. OmniObject3D: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 803–814, 2023.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21469–21480, 2025.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14419–14429, 2026.

Tianhao Zhao, Youjia Zhang, Hang Long, Jinshen Zhang, Wenbing Li, Yang Yang, Gongbo Zhang, Jozef Hladký, Matthias Nießner, and Wei Yang. LATO: 3d mesh flow matching with structured TOpology preserving LAtents. In *Forty-third International Conference on Machine Learning*, 2026.

Zhongtian Zheng, Xifeng Gao, Zherong Pan, Wei Li, Peng-Shuai Wang, Guoping Wang, and Kui Wu. Visual-preserving mesh repair. *IEEE Transactions on Visualization and Computer Graphics*, 30(9):6586–6597, 2024.

A Additional Experimental Results

A.1 Absolute Irregularity Counts

The main paper reports normalized topology-irregularity ratios to account for differences in mesh complexity across objects and benchmarks. For completeness, Table 4 reports the corresponding absolute numbers of non-manifold edges (NME), non-manifold vertices (NMV), and boundary edges (BND). All statistics are computed on the same unrepaired raw indexed meshes used in the main topology evaluation. For learned methods, we report mean $\pm$ standard deviation over three independent training runs. The absolute counts exhibit the same overall trend as the normalized ratios in Table 1, with token-level TopoFT achieving the lowest mean irregularity counts across all four benchmarks.

Table 4: Absolute mesh-topology irregularity counts. We report the average numbers of non-manifold edges (NME), non-manifold vertices (NMV), and boundary edges (BND) per generated mesh. Learned methods are reported as mean $\pm$ standard deviation over three training seeds.

Dataset	Method	NME Count ↓	NMV Count ↓	BND Count ↓
GSO	TRELLIS.2	20,882	434	17,221
	DSO	$14,710 \pm 684$	220 ± 9	$15,517 \pm 466$
	TopoFT(mesh)	$8,444 \pm 86$	133 ± 13	$11,056 \pm 495$
	TopoFT(token)	$\mathbf{5,817 \pm 641}$	$\mathbf{47 \pm 17}$	$\mathbf{9,178 \pm 696}$
Fusion 360	TRELLIS.2	9,175	94	6,557
	DSO	$5,984 \pm 441$	68 ± 6	$6,659 \pm 408$
	TopoFT(mesh)	$3,708 \pm 60$	43 ± 3	$3,876 \pm 149$
	TopoFT(token)	$\mathbf{2,883 \pm 318}$	$\mathbf{15 \pm 3}$	$\mathbf{3,313 \pm 315}$
Omni Object3D	TRELLIS.2	52,756	1,023	34,398
	DSO	$25,427 \pm 1,104$	389 ± 14	$25,166 \pm 1,005$
	TopoFT(mesh)	$12,377 \pm 239$	179 ± 4	$15,362 \pm 517$
	TopoFT(token)	$\mathbf{8,479 \pm 1,122}$	$\mathbf{40 \pm 21}$	$\mathbf{11,094 \pm 1,841}$
Toys4K	TRELLIS.2	40,539	802	21,020
	DSO	$23,460 \pm 1,342$	511 ± 33	$20,037 \pm 494$
	TopoFT(mesh)	$16,368 \pm 415$	419 ± 14	$15,580 \pm 302$
	TopoFT(token)	$\mathbf{8,501 \pm 1,825}$	$\mathbf{111 \pm 63}$	$\mathbf{11,005 \pm 1,040}$

A.2 Mesh Complexity Statistics

Table 5 reports mesh-complexity ratios relative to the corresponding pretrained TRELLIS.2 outputs. Following the notation used in the complexity-preserving regularization, we denote the vertex, edge, and face sets of a generated mesh by $\mathcal{V}$, $\mathcal{E}$, and $\mathcal{F}$. For evaluation, the corresponding pretrained TRELLIS.2 output is used as the reference, with sets $\mathcal{V}^{\mathrm{ref}}$, $\mathcal{E}^{\mathrm{ref}}$, and $\mathcal{F}^{\mathrm{ref}}$. We report

$$R_V = \frac{|\mathcal{V}|}{|\mathcal{V}^{\mathrm{ref}}|}, \qquad R_E = \frac{|\mathcal{E}|}{|\mathcal{E}^{\mathrm{ref}}|}, \qquad R_F = \frac{|\mathcal{F}|}{|\mathcal{F}^{\mathrm{ref}}|}. \tag{17}$$

In particular, R_E corresponds directly to the edge-count ratio used by the complexity-preserving regularizer in Eq. 4. Ratios close to 1.0 indicate that the mesh complexity of the pretrained generator is largely preserved. These results indicate that the topology improvements are not accompanied by substantial reductions in average mesh complexity.

A.3 Effect of Complexity Regularization

We further examine the role of the complexity-preserving regularization in TopoFT. Specifically, we remove the complexity-preserving regularization term from the topology objective while keeping the remaining training settings unchanged. On the validation set, we track the previously defined R_V, R_E, and R_F, together with the fraction of empty generated meshes.

As shown in Fig. 4, removing the complexity-preserving regularization leads to a consistent reduction in mesh complexity across all three training seeds. The vertex, edge, and face ratios decrease

Table 5: Mesh complexity relative to the corresponding pretrained TRELLIS.2 outputs. R_V, R_E, and R_F denote the vertex-, edge-, and face-count ratios with respect to the pretrained reference meshes. Learned methods are reported as mean $\pm$ standard deviation over three training seeds. Values closer to 1.0 indicate better preservation of the pretrained mesh complexity.

Dataset	Method	R_V	R_E	R_F
GSO	DSO	0.977 ± 0.009	0.973 ± 0.009	0.972 ± 0.009
	TopoFT(mesh)	$\mathbf{0.986 \pm 0.002}$	$\mathbf{0.981 \pm 0.002}$	0.979 ± 0.002
	TopoFT(token)	1.027 ± 0.037	1.021 ± 0.037	$\mathbf{1.019 \pm 0.037}$
Fusion 360	DSO	0.986 ± 0.011	0.982 ± 0.012	0.981 ± 0.012
	TopoFT(mesh)	$\mathbf{0.993 \pm 0.002}$	$\mathbf{0.990 \pm 0.002}$	$\mathbf{0.989 \pm 0.002}$
	TopoFT(token)	1.049 ± 0.053	1.045 ± 0.053	1.043 ± 0.053
Omni Object3D	DSO	0.985 ± 0.006	0.976 ± 0.007	0.971 ± 0.007
	TopoFT(mesh)	0.984 ± 0.001	0.973 ± 0.001	0.968 ± 0.001
	TopoFT(token)	$\mathbf{0.997 \pm 0.010}$	$\mathbf{0.985 \pm 0.010}$	$\mathbf{0.979 \pm 0.011}$
Toys4K	DSO	0.978 ± 0.012	0.967 ± 0.013	0.961 ± 0.013
	TopoFT(mesh)	0.988 ± 0.001	0.976 ± 0.001	0.968 ± 0.001
	TopoFT(token)	$\mathbf{1.001 \pm 0.019}$	$\mathbf{0.984 \pm 0.018}$	$\mathbf{0.975 \pm 0.018}$

progressively during optimization and approach zero at later training steps. This degeneration is accompanied by a sharp increase in the fraction of empty meshes. The closely aligned trajectories across the three seeds indicate that this behavior is consistent across training runs.

These results show that optimizing the normalized irregularity ratios without controlling mesh complexity can lead to a degenerate solution in which the model reduces topology defects by progressively removing geometric elements rather than correcting irregular connectivity. The complexity-preserving regularization prevents this collapse by discouraging substantial deviations in mesh complexity from the pretrained generator.

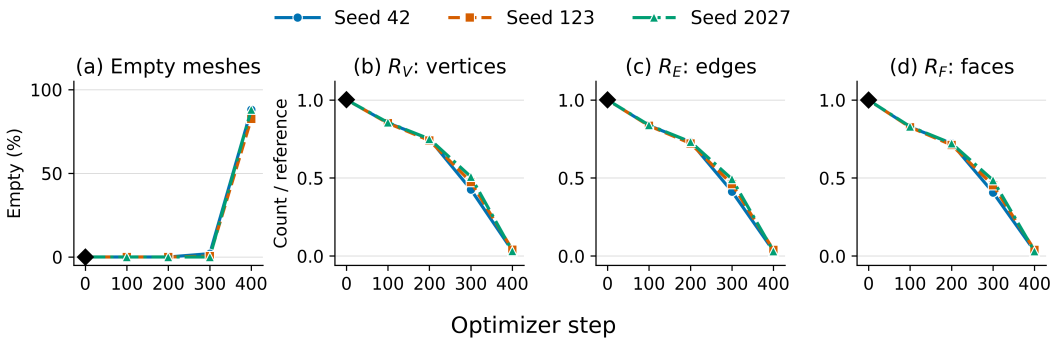

Figure 4: **Effect of removing complexity regularization.** We disable the mesh-complexity term and track training . (a) Fraction of empty generated meshes. (b–d) Vertex, edge, and face counts relative to the reference meshes. Without complexity regularization, all three complexity ratios progressively decrease and eventually approach zero, while the fraction of empty meshes increases sharply. The consistent behavior across seeds indicates a systematic complexity-collapse failure mode.

A.4 Complete Geometric Fidelity Results

We report the complete geometric fidelity results for all compared methods in Table 6. The evaluation protocol and metric definitions are provided in Appendix B.4.1.

Table 6: **Geometric fidelity.** We report F-scores at three distance thresholds. Learned methods are reported as mean ± standard deviation over three training seeds. The best mean before rounding on each benchmark is shown in bold.

Dataset	Method	Chamfer ($\times 10^{-2}$) ↓	P95 ($\times 10^{-2}$) ↓	F@0.005 (%) ↑	F@0.01 (%) ↑	F@0.02 (%) ↑
GSO	TRELLIS.2	6.02	31.82	1.67	6.34	14.22
	DSO	6.00 ± 0.00	31.82 ± 0.01	1.67 ± 0.00	6.35 ± 0.01	14.28 ± 0.02
	TopoFT(mesh)	5.99 ± 0.00	31.80 ± 0.01	1.68 ± 0.00	6.39 ± 0.01	14.32 ± 0.01
	TopoFT(token)	$\mathbf{5.96 \pm 0.03}$	$\mathbf{31.74 \pm 0.05}$	$\mathbf{1.69 \pm 0.00}$	$\mathbf{6.51 \pm 0.06}$	$\mathbf{14.53 \pm 0.12}$
Fusion 360	TRELLIS.2	9.03	36.87	2.31	6.27	12.49
	DSO	9.03 ± 0.00	36.86 ± 0.00	2.32 ± 0.01	6.29 ± 0.01	12.50 ± 0.01
	TopoFT(mesh)	9.02 ± 0.00	36.84 ± 0.01	2.35 ± 0.01	6.35 ± 0.03	12.59 ± 0.02
	TopoFT(token)	$\mathbf{8.98 \pm 0.03}$	$\mathbf{36.76 \pm 0.06}$	$\mathbf{2.42 \pm 0.04}$	$\mathbf{6.48 \pm 0.07}$	$\mathbf{12.76 \pm 0.11}$
Omni Object3D	TRELLIS.2	4.90	28.02	1.54	6.20	15.00
	DSO	4.88 ± 0.00	28.01 ± 0.00	1.54 ± 0.00	6.22 ± 0.01	15.02 ± 0.01
	TopoFT(mesh)	4.88 ± 0.00	28.00 ± 0.01	$\mathbf{1.54 \pm 0.00}$	6.23 ± 0.00	15.06 ± 0.01
	TopoFT(token)	$\mathbf{4.87 \pm 0.01}$	$\mathbf{28.00 \pm 0.00}$	1.54 ± 0.00	$\mathbf{6.26 \pm 0.01}$	$\mathbf{15.11 \pm 0.02}$
Toys4K	TRELLIS.2	6.22	31.90	1.44	5.78	14.11
	DSO	6.22 ± 0.00	31.91 ± 0.01	1.44 ± 0.00	5.81 ± 0.01	14.14 ± 0.02
	TopoFT(mesh)	6.21 ± 0.00	31.88 ± 0.00	1.44 ± 0.00	5.84 ± 0.00	14.20 ± 0.01
	TopoFT(token)	$\mathbf{6.19 \pm 0.01}$	$\mathbf{31.84 \pm 0.02}$	$\mathbf{1.45 \pm 0.01}$	$\mathbf{5.91 \pm 0.03}$	$\mathbf{14.39 \pm 0.09}$

Table 7: "Target reached" denotes the fraction of completed QEM runs whose final face count does not exceed the 10% target. Face ratio denotes the mean final-to-original face-count ratio over completed QEM runs, with values closer to 10% being better. The best result on each benchmark is shown in bold.

Dataset	Method	Reached (%) ↑	Face ratio (%)
GSO	TRELLIS.2	73.90	12.23
	DSO	82.97 ± 1.03	10.99 ± 0.06
	TopoFT(mesh)	93.77 ± 0.81	10.39 ± 0.03
	TopoFT(token)	$\mathbf{98.83 \pm 0.90}$	$\mathbf{10.05 \pm 0.04}$
Fusion 360	TRELLIS.2	90.98	11.22
	DSO	94.92 ± 0.44	10.48 ± 0.03
	TopoFT(mesh)	98.21 ± 0.10	10.17 ± 0.01
	TopoFT(token)	$\mathbf{99.44 \pm 0.31}$	$\mathbf{10.05 \pm 0.03}$
Omni Object3D	TRELLIS.2	52.68	14.70
	DSO	78.11 ± 2.01	11.25 ± 0.10
	TopoFT(mesh)	95.69 ± 0.10	10.30 ± 0.01
	TopoFT(token)	$\mathbf{99.33 \pm 0.75}$	$\mathbf{10.04 \pm 0.05}$
Toys4K	TRELLIS.2	62.09	14.79
	DSO	73.14 ± 1.35	12.45 ± 0.11
	TopoFT(mesh)	83.83 ± 0.60	11.36 ± 0.05
	TopoFT(token)	$\mathbf{95.75 \pm 3.48}$	$\mathbf{10.25 \pm 0.27}$

A.5 Complete Downstream Mesh Processing Results

Tables 7 and 8 report the complete downstream results for QEM simplification and CuraEngine slicing across all four benchmarks. The full results are consistent with the trends summarized in the main paper: TopoFT improves the reliability of QEM simplification and generally reduces slicing-related open-wall artifacts. The token-level variant achieves the strongest overall downstream performance, while the mesh-level variant also consistently improves over the pretrained model and DSO.

Table 8: **Cura slicing results.** Valid denotes the fraction of all benchmark objects scheduled for evaluation that produce valid, non-empty model toolpaths. OCF denotes the fraction of valid-toolpath meshes that are open-contour-free. OWR denotes the open-wall ratio under a $0.1\,\mathrm{mm}$ gap threshold between the endpoints of the same model wall-contour path. AL denotes the affected-layer ratio, evaluated over layers containing at least one model wall-contour path. OWR and AL are computed per mesh and averaged over meshes with valid, non-empty model toolpaths. Learned methods are reported as mean $\pm$ standard deviation over three training seeds. The best mean on each benchmark is shown in bold.

Dataset	Method	Valid (%) ↑	OCF (%) ↑	OWR (%) ↓	AL (%) ↓
GSO	TRELLIS.2	99.00	46.46	22.92	49.35
	DSO	99.47 ± 0.15	49.87 ± 0.71	20.33 ± 0.21	47.22 ± 0.34
	TopoFT(mesh)	99.53 ± 0.23	$\mathbf{53.61 \pm 1.44}$	17.63 ± 0.36	41.60 ± 0.70
	TopoFT(token)	$\mathbf{99.80 \pm 0.00}$	50.50 ± 4.08	$\mathbf{14.46 \pm 1.22}$	$\mathbf{36.17 \pm 2.57}$
Fusion 360	TRELLIS.2	98.92	54.82	16.65	29.68
	DSO	98.85 ± 0.06	56.58 ± 0.48	15.13 ± 0.26	28.93 ± 0.56
	TopoFT(mesh)	99.21 ± 0.15	61.88 ± 0.27	12.01 ± 0.04	23.50 ± 0.18
	TopoFT(token)	$\mathbf{99.43 \pm 0.15}$	$\mathbf{64.12 \pm 1.53}$	$\mathbf{9.43 \pm 0.50}$	$\mathbf{19.94 \pm 0.52}$
Omni Object3D	TRELLIS.2	98.19	54.79	15.65	30.41
	DSO	99.70 ± 0.00	61.06 ± 0.75	13.79 ± 0.25	28.99 ± 0.55
	TopoFT(mesh)	$\mathbf{99.71 \pm 0.02}$	64.90 ± 0.91	10.97 ± 0.19	23.93 ± 0.55
	TopoFT(token)	$\mathbf{99.71 \pm 0.19}$	$\mathbf{65.07 \pm 1.24}$	$\mathbf{7.98 \pm 1.12}$	$\mathbf{18.96 \pm 2.07}$
Toys4K	TRELLIS.2	98.20	46.83	21.90	41.34
	DSO	98.78 ± 0.17	48.79 ± 0.39	18.90 ± 0.33	39.64 ± 0.40
	TopoFT(mesh)	98.87 ± 0.17	54.12 ± 0.95	16.59 ± 0.32	35.26 ± 0.58
	TopoFT(token)	$\mathbf{99.40 \pm 0.34}$	$\mathbf{59.04 \pm 1.47}$	$\mathbf{11.83 \pm 1.91}$	$\mathbf{29.12 \pm 2.97}$

A.6 Qualitative Comparisons

Figures 5 and 6 provide complementary qualitative comparisons between pretrained TRELLIS.2 and TopoFT.

Topology visualization. Both methods use the same viewing configurations and deterministic marker-selection procedure. For legibility, candidates of each defect type are sorted by their mesh edge or vertex indices and selected at evenly spaced indices in the sorted sequence when subsampling is needed.

A.7 Sensitivity to Topology-Objective Weights

We examine the sensitivity of TopoFT to the relative weights (w_1, w_2, w_3) of the three topology terms on GSO. We evaluate four weight configurations using a fixed training seed while keeping all other training and evaluation settings unchanged. As shown in Table 9, TopoFT achieves low topology irregularity across all tested configurations, while the exact NME, NMV, and BND values depend on the relative weighting. In particular, $(0.4, 0.2, 0.4)$ yields the lowest values for all three metrics in this study, whereas NMV shows relatively larger variation across configurations due to its small absolute magnitude. **For the main experiments, we use the fixed configuration $(0.3, 0.2, 0.5)$ across all datasets rather than tuning the weights for each benchmark**. Overall, the results indicate that the topology-objective weights affect the resulting topology profile, while TopoFT remains effective across the tested configurations.

B Additional Experimental Details

This appendix provides supplementary details for the topology diagnostic study, training configuration, baseline construction, validation procedure, and evaluation protocols used in our experiments.

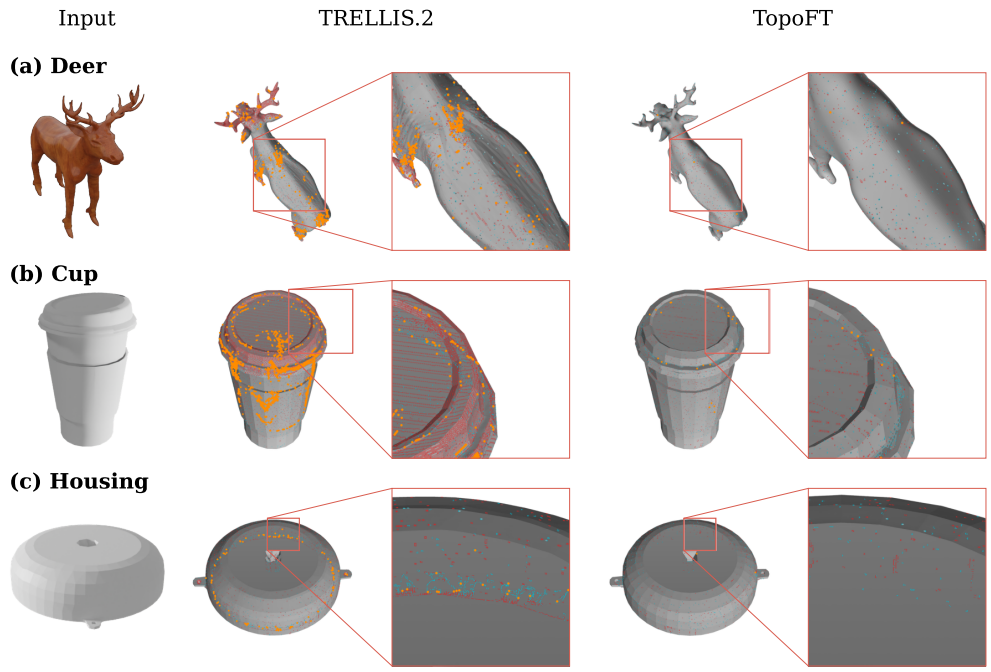

Figure 5: **Qualitative comparison of mesh topology.** Non-manifold edges (NME) are drawn as red line segments and boundary edges (BND) as cyan line segments along their projected mesh edges. Non-manifold vertices (NMV) are shown as orange dots at their projected vertex positions. Light-red boxes and connectors indicate the regions shown in the enlarged views. TopoFT substantially reduces localized topology irregularities.

Table 9: **Sensitivity to topology-objective weights.** We vary (w_1, w_2, w_3) while keeping all other settings fixed. The configuration used in the main experiments is marked by †.

(w_1, w_2, w_3)	NME (‰) ↓	NMV (‰) ↓	BND (‰) ↓
$(1/3, 1/3, 1/3)$	3.82	0.16	6.46
$(0.3, 0.3, 0.4)$	4.32	0.22	6.34
$(0.4, 0.2, 0.4)$	**3.43**	**0.07**	**5.25**
$(0.3, 0.2, 0.5)^{\dagger}$	3.99	0.17	5.80

B.1 Topology Diagnostic Protocol

We provide additional details for the controlled diagnostic study in Sec. 2.2. The study uses 5,000 topology-regular meshes from the Fusion 360 Gallery Segmentation Dataset and progressively introduces components of the pretrained TRELLIS.2 geometry pipeline. No hole filling, vertex welding, remeshing, mesh repair, or connected-component filtering is applied before evaluation.

Diagnostic set. We select 5,000 triangular meshes from the official Fusion 360 training split that satisfy strict mesh-validity checks and contain no non-manifold edges, non-manifold vertices, or boundary edges.

Diagnostic pipelines. Following Fig. 2, we evaluate four progressively more complete pipelines with O-Voxel decoding and mesh extraction at resolution 512.

Pipeline A performs a direct O-Voxel round trip by converting each source mesh to O-Voxel and extracting it back to a mesh. Pipeline B additionally performs an SC-VAE encode–decode reconstruction of the O-Voxel representation. The SC-VAE encoder is used only for this diagnostic recon-

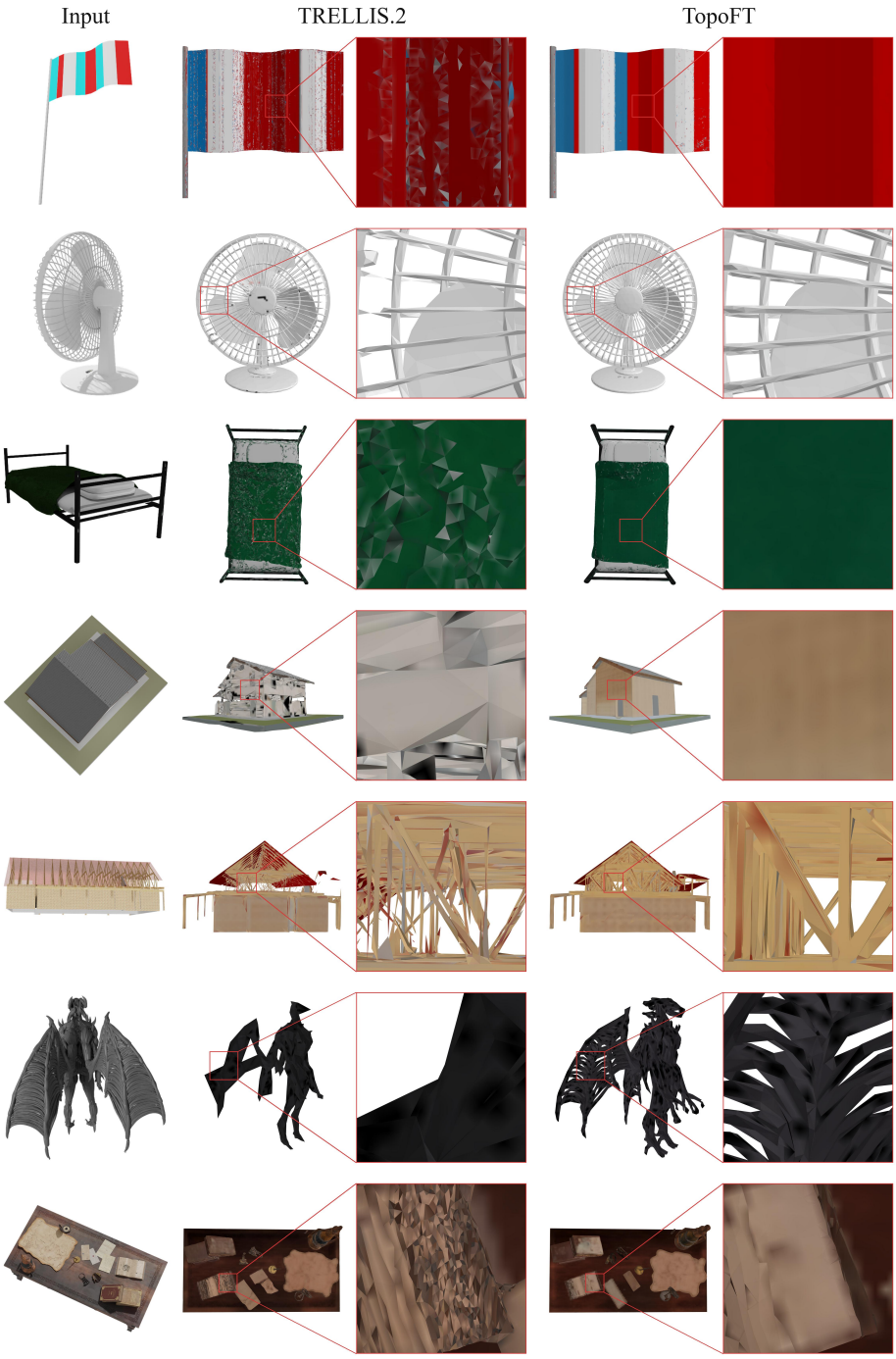

Figure 6: **Qualitative comparison of rendered appearance.** TopoFT largely preserves the overall appearance and local surface structure of pretrained TRELLIS.2.

struction and is not part of the forward generation pipeline. The comparison between Pipelines A and B therefore measures the additional topology irregularity introduced by SC-VAE reconstruction.

Pipeline C introduces Stage-2 shape-SLat generation while retaining an oracle spatial structure derived from the paired source mesh. The active spatial locations $\mathcal{S}$ are obtained from the source mesh, while the Stage-2 geometry flow model generates the corresponding shape-SLat tokens x_0.

Table 10: **Stage-wise topology diagnosis on 5,000 topology-regular Fusion 360 meshes.** Pipelines A–D correspond to the four diagnostic pipelines in Fig. 2. "Irregular meshes" denotes the fraction of outputs containing at least one NME, NMV, or BND. The three rightmost columns report the mean per-mesh topology ratios defined in Sec. 2.

Pipeline	Configuration	Irregular meshes (‰)	$\overline{\rho}_{\mathrm{NME}}$ (‰)	$\overline{\rho}_{\mathrm{NMV}}$ (‰)	$\overline{\rho}_{\mathrm{BND}}$ (‰)
A	O-Voxel meshing	518.8	0.220	0.002	0.381
B	SC-VAE reconstruction	616.0	0.256	0.003	0.380
C	Stage-2 shape-SLat generation	907.0	6.104	0.133	4.740
D	Full TRELLIS.2	933.6	8.111	0.215	7.712

Pipelines B and C therefore use spatial locations derived from the same source mesh but differ in how the shape-SLat features are obtained. Their difference isolates the topology change associated with Stage-2 shape-SLat generation without introducing Stage-1 prediction errors.

Pipeline D corresponds to full pretrained TRELLIS.2 generation. It replaces the oracle spatial structure in Pipeline C with the structure predicted by Stage-1. The comparison between Pipelines C and D therefore measures the additional topology change associated with Stage-1 structure prediction.

Complete diagnostic results. The complete results confirm the trend shown in Fig. 2. O-Voxel meshing already introduces topology irregularity into some topology-regular inputs, while SC-VAE reconstruction causes only small changes in the mean irregularity ratios. The largest increase occurs when Stage-2-generated shape-SLat features are introduced. The mean ρ_{NME}, ρ_{NMV}, and ρ_{BND} increase by approximately $23.8\times$, $44.3\times$, and $12.5\times$, while the fraction of irregular meshes increases from 61.60% to 90.70%. Replacing the oracle spatial structure with the Stage-1 prediction further increases the irregular-mesh rate to 93.36%. This additional increase is substantially smaller than that associated with Stage-2 shape-SLat generation.

B.2 Training and Validation Details

Data and preprocessing. The training and validation partitions follow Sec. 4.1. The 30,000 training images form the available conditioning pool from which conditioning views are selected during post-training. All rendered conditioning images are processed using the TRELLIS.2 training preprocessing pipeline and fixed before post-training. No random image augmentation is applied.

For each conditioning view, the frozen Stage-1 sparse-structure model is evaluated once before post-training and the resulting sparse structure is cached. The same cached structure is reused throughout training and validation. This isolates Stage-2 optimization from variations in Stage-1 sampling.

Training configuration. Table 11 summarizes the optimization settings used by the learned methods.

Stage-2 sampling. For each selected conditioning image, TopoFT generates $K = 8$ shape-SLat samples from the current Stage-2 model. The samples share the same conditioning representation and cached Stage-1 sparse structure but use independent Stage-2 sampling noise. We use the `FlowEulerGuidanceIntervalSampler` with 12 sampling steps. Classifier-free guidance uses a strength of 7.5 and a guidance rescale of 0.5.

The `logit-normal` distribution $p(t)$ of flow-matching timesteps. For each generated sample, the flow-matching timestep is independently sampled as

$$z \sim \mathcal{N}(0,1), \qquad t = \mathrm{sigmoid}(1+z), \tag{18}$$

corresponding to a `logit-normal` distribution (Xiang et al., 2025) with location parameter 1 and scale parameter 1. The sampled timestep is used to construct the noisy state and target velocity in Sec. 3.3.

Optimization. The effective training batch size of 24 counts generated Stage-2 shape-SLat samples rather than conditioning images or individual tokens. With $K = 8$ samples per conditioning

Table 11: Training configuration used for the learned methods unless otherwise specified.

Setting	Value
Stage-2 samples per conditioning visit	$K = 8$
Stage-2 sampling steps	12
CFG strength	7.5
CFG rescale	0.5
w_1, w_2, w_3, w_4	$0.3, 0.2, 0.5, 0.1$
ζ	1×10^{-12}
Fine-tuned module	Stage-2 geometry flow
Frozen modules	Image encoder, Stage-1, shape decoder
LoRA rank	64
LoRA α	128
LoRA dropout	0
Optimizer	AdamW
Learning rate	1×10^{-5}
Weight decay	0.01
Effective training batch size	24
Optimizer steps per online round	10
Validation interval	100 updates
Training seeds	42, 123, 2027
Maximum optimizer updates	3,500

visit, one effective batch therefore contains samples from three conditioning-image visits. We optimize the LoRA parameters using AdamW with learning rate 1×10^{-5} and weight decay 0.01. LoRA uses rank 64, scaling parameter $\alpha = 128$, and zero dropout. LoRA is applied to the self-attention and cross-attention projection layers in all Transformer blocks of the Stage-2 model, while the remaining parameters are kept frozen.

Online optimization schedule. At the beginning of each online round, the current Stage-2 model is frozen as the sampling model for that round. The collected samples are partitioned into mini-batches and each sample is used once for optimization. Each online round performs 10 optimizer updates before the sampling model is refreshed.

Complexity references. For each conditioning view used during training or validation, we generate one fixed complexity reference using the frozen pretrained TRELLIS.2 model. The reference is generated using the same conditioning image and cached Stage-1 sparse structure as the corresponding training sample. The resulting references are fixed before post-training and shared by TopoFT, DSO, all training seeds, and the corresponding ablations. The reference statistics and token-level decomposition follow the definitions in the main method section.

Degenerate decoded meshes. A decoded mesh is treated as empty when it contains no triangular faces, including both meshes with no vertices and meshes containing vertices but no faces. For such outputs, the normalized NME, NMV, and BND ratios are undefined; we therefore assign the maximal penalty to each topology term. Since an empty mesh has no edges, its complexity-preserving penalty equals w_4, giving

$$D(\mathcal{M}_{\text{empty}}) = w_1 + w_2 + w_3 + w_4 = 1.1. \tag{19}$$

Empty outputs are retained in the K-sample candidate group rather than discarded. Their cost is distributed over the token support of the cached reference mesh such that the token-wise contributions preserve the corresponding mesh-level cost, after which the same group-relative advantage computation is applied as for non-empty candidates.

Validation and checkpoint selection. For each object in the validation set, we generate one sample using a fixed conditioning image, cached Stage-1 sparse structure, and Stage-2 generation seed shared across methods, training runs, and checkpoints. The regularized topology cost is computed for each sample and averaged over the validation set. All learned methods are evaluated at the same

optimizer-update intervals. For each training run, we select the checkpoint with the lowest mean validation cost within 3,500 optimizer updates.

B.3 DSO Baseline Construction

We adapt the direct reward optimization formulation of DSO (Li et al., 2025) to topology feedback. Unlike TopoFT, which continually regenerates samples from the evolving Stage-2 model, DSO trains on an offline dataset generated once from the frozen pretrained TRELLIS.2 model. For each selected conditioning image, the pretrained Stage-2 geometry flow model generates $K = 8$ shape-SLat samples, which are decoded and evaluated using the same regularized topology cost and per-view complexity references as TopoFT. The resulting offline corpus contains 84,000 Stage-2 samples, corresponding exactly to 3,500 optimizer updates with an effective batch size of 24. Each sample is used once during training, with no sample reuse.

The topology cost is continuous, whereas DSO assumes a binary reward signal. We therefore convert the topology cost into a binary reward using a global threshold over the offline training dataset. Let $D^{(i)}$ denote the regularized topology cost of offline sample i. For each DSO run, we define

$$\tau_{\mathrm{DSO}} = \mathrm{median}_{i \in \mathcal{D}_{\mathrm{offline}}} D^{(i)}, \tag{20}$$

and assign

$$r^{(i)} = \begin{cases} 1, & D^{(i)} \le \tau_{\mathrm{DSO}}, \\ 0, & D^{(i)} > \tau_{\mathrm{DSO}}. \end{cases} \tag{21}$$

The threshold is computed only from the offline training dataset. Using a dataset-level threshold provides a common topology-quality criterion across conditioning inputs rather than imposing a separate ranking within each sample group.

Following the DRO objective of DSO (Li et al., 2025), the binary reward determines the sign of the flow-matching objective:

$$\mathcal{L}_{\mathrm{DSO}} = \frac{1}{N} \sum_{j=1}^{N} \left(2r^{(i)} - 1\right) \left\| v_{\theta,j}^{(i)} - v_j^{\star,(i)} \right\|_2^2. \tag{22}$$

Here, t denotes the sampled flow-matching timestep, and the dependence of $v_{\theta,j}^{(i)}$ on t through the interpolated state $x_t^{(i)}$ is omitted for notational simplicity. All remaining optimization settings follow Table 11. For the multi-seed evaluation, each DSO run independently generates its offline dataset using the corresponding random seed and therefore differs in both offline sample generation and optimization randomness.

B.4 Evaluation Protocols

B.4.1 Geometric Fidelity

We evaluate geometric fidelity on GSO, Fusion 360, OmniObject3D, and Toys4K. For each generated mesh, we sample surface points by area to match the number of points in the corresponding ground-truth point cloud. This results in 16,384 prediction points for GSO, Fusion 360, and OmniObject3D, and 10,000 points for Toys4K.

For GSO, the ground-truth point clouds are obtained by offline surface sampling from the official meshes. For Fusion 360, we use offline surface samples from the official s2.0.1 test meshes stored in a validated cache. For OmniObject3D, we directly use the official 16,384-point PLY files. For Toys4K, we directly use the provided 10,000-point NPZ point clouds without resampling.

Before distance computation, the predicted surface samples and ground-truth points are normalized independently for each object. For a point set X, we subtract the center of its axis-aligned bounding box and divide by its maximum bounding-box extent. No axis permutation, sign flip, rotation, rigid registration, or ICP alignment is applied.

Let P denote the normalized prediction point set and Q the normalized ground-truth point set. We define the directional nearest-neighbor distances as

$$d_{P\to Q}(p) = \min_{q\in Q} \|p-q\|_2, \qquad d_{Q\to P}(q) = \min_{p\in P} \|q-p\|_2. \tag{23}$$

The symmetric squared Chamfer distance is

$$\mathrm{CD}(P,Q) = \frac{1}{|P|}\sum_{p\in P} d_{P\to Q}(p)^2 + \frac{1}{|Q|}\sum_{q\in Q} d_{Q\to P}(q)^2. \tag{24}$$

To characterize larger surface deviations, we report a bidirectional P95 defined as the larger of the two directional 95th-percentile distances:

$$\mathrm{P95}(P,Q) = \max\left(\mathrm{Quantile}_{0.95}\left(\{d_{P\to Q}(p)\}_{p\in P}\right), \mathrm{Quantile}_{0.95}\left(\{d_{Q\to P}(q)\}_{q\in Q}\right)\right). \tag{25}$$

For a distance threshold τ, precision and recall are defined as

$$\mathrm{Prec}_\tau(P,Q) = \frac{1}{|P|}\sum_{p\in P} \mathbb{I}\left[d_{P\to Q}(p)\le\tau\right], \tag{26}$$

and

$$\mathrm{Rec}_\tau(P,Q) = \frac{1}{|Q|}\sum_{q\in Q} \mathbb{I}\left[d_{Q\to P}(q)\le\tau\right]. \tag{27}$$

The corresponding F-score is

$$F_\tau = \frac{2\,\mathrm{Prec}_\tau\,\mathrm{Rec}_\tau}{\mathrm{Prec}_\tau + \mathrm{Rec}_\tau}, \tag{28}$$

with $F_\tau = 0$ when both precision and recall are zero.

We evaluate F-score at $\tau \in \{0.005, 0.01, 0.02\}$. Because prediction and ground-truth point sets are independently normalized before evaluation, these thresholds are expressed in normalized coordinates and do not correspond to physical units.

Surface sampling is deterministic under the prescribed evaluation seed. For a given benchmark object, the same prediction-sampling seed is used across methods and training seeds, ensuring paired geometric evaluation.

For each method and evaluation run, each geometric metric is computed independently for each object and then averaged within each benchmark over objects for which that metric is finite and successfully computed. Objects with failed or non-finite results for a metric are excluded from the average of that metric. For learned methods, we first compute one benchmark mean for each training seed in $\{42, 123, 2027\}$, and report the mean and sample standard deviation across these three runs.

Mesh-complexity statistics are computed separately on the original raw indexed meshes used for topology evaluation, without vertex welding, remeshing, simplification, or topology repair.

B.4.2 Downstream Mesh Processing

All downstream experiments operate directly on the generated meshes without topology repair. All compared methods use identical downstream configurations.

QEM simplification. We simplify each raw indexed mesh independently using PyMeshLab with quadric edge-collapse simplification. For an input mesh with F faces, the requested target face count is

$$F_{\mathrm{target}} = \max\left(1, \lceil 0.1F \rceil\right). \tag{29}$$

Boundary preservation and topology preservation are enabled, and optimal vertex placement is used. The same simplification configuration is applied to all methods.

A sample is considered to reach the requested simplification target when

$$F_{\mathrm{final}} \le F_{\mathrm{target}}. \tag{30}$$

The target-reaching rate is computed over completed QEM runs. If simplification raises an error or returns an invalid mesh, the sample is marked as a QEM failure and excluded because no valid final face count is available. The mean final-to-original face-count ratio F_{final}/F is likewise computed only over completed QEM runs. Values closer to 10% indicate that the requested simplification level is more closely achieved.

Cura slicing. We slice meshes using CuraEngine with a fixed Creality Ender-3 machine profile shared by all methods. To isolate generator-side topology from slicer-side correction, we disable CuraEngine operations that modify or repair mesh connectivity. All remaining slicing settings are kept identical across methods. The layer height is fixed to $0.2\,\text{mm}$, with the initial layer height also set to $0.2\,\text{mm}$. Each mesh is uniformly scaled such that its longest axis-aligned bounding-box extent is $80\,\text{mm}$. The scaled mesh is centered on the build plate and translated along the z axis such that its minimum z coordinate is zero. The original orientation is preserved, with no rotation or automatic reorientation.

Open-wall detection. For each slicing layer containing at least one model wall-contour path, we inspect the start and end points of each path. Endpoint matching is restricted to the two endpoints of the same path; endpoints from different paths are not matched. A path is considered closed when the Euclidean distance between its endpoints does not exceed

$$\tau_{\text{gap}} = 0.1\,\text{mm}. \tag{31}$$

Otherwise, both endpoints are classified as open-wall endpoints and the path is classified as an open-wall path. A layer containing at least one open-wall path is classified as an affected layer.

Evaluation metrics. A mesh produces a valid toolpath when CuraEngine completes slicing successfully and generates a non-empty model toolpath. For each benchmark and evaluation run, the denominator of the valid-toolpath rate is the full set of benchmark objects scheduled for evaluation, with one requested mesh per object. Missing or failed mesh generations, slicing failures, and empty model toolpaths count as invalid. The open-contour-free rate (OCF) is the fraction of meshes with valid toolpaths for which no open-wall path is detected.

The open-wall ratio (OWR) is

$$\text{OWR} = \frac{N_{\text{open-wall paths}}}{N_{\text{contour paths}}}, \tag{32}$$

and the affected-layer ratio (AL) is

$$\text{AL} = \frac{N_{\text{affected layers}}}{N_{\text{nonempty layers}}}. \tag{33}$$

For each mesh, $N_{\text{contour paths}}$ counts all model wall-contour paths, and $N_{\text{nonempty layers}}$ counts layers containing at least one such path. Both OWR and AL are computed per mesh over meshes with valid, non-empty model toolpaths and then averaged within each benchmark, so that each included mesh contributes equally to the reported metric. All downstream metrics are computed separately for each benchmark under the same evaluation protocol. For DSO and both TopoFT variants, each benchmark-level metric is first computed separately for each training seed, and the mean and standard deviation are then reported across the three seeds.

C Proof of Proposition 1

Property (i): Exact global-to-local preservation. Let $\mathcal{M}^{(i)} = \text{Dec}(x_0^{(i)})$ denote the decoded mesh of sample i, with vertex and edge sets $\mathcal{V}^{(i)}$ and $\mathcal{E}^{(i)}$. From Eq. 8,

$$\sum_j C_j(\mathcal{V}') = |\mathcal{V}'|, \qquad \sum_j C_j(\mathcal{E}') = |\mathcal{E}'|.$$

Therefore, summing Eq. 9 over all tokens gives

$$\sum_j q_j^{(i)} = w_1 \rho_{\text{NME}}(\mathcal{M}^{(i)}) + w_2 \rho_{\text{NMV}}(\mathcal{M}^{(i)}) + w_3 \rho_{\text{BND}}(\mathcal{M}^{(i)}) + w_4 s^{(i)} \frac{|\mathcal{E}^{\text{ref}}| - |\mathcal{E}^{(i)}|}{|\mathcal{E}^{\text{ref}}|}.$$

Since

$$s^{(i)} = \operatorname{sign}\left(|\mathcal{E}^{\mathrm{ref}}| - |\mathcal{E}^{(i)}|\right)$$

and $\operatorname{sign}(x)x = |x|$, the last term becomes

$$w_4 \left| 1 - \frac{|\mathcal{E}^{(i)}|}{|\mathcal{E}^{\mathrm{ref}}|} \right|.$$

Hence,

$$\sum_j q_j^{(i)} = D^{(i)}.$$

Moreover,

$$\sum_j \bar{q}_j = \frac{1}{K} \sum_{k=1}^{K} \sum_j q_j^{(k)} = \frac{1}{K} \sum_{k=1}^{K} D^{(k)} = \bar{D}.$$

Therefore,

$$\sum_j a_j^{(i)} = \frac{\sum_j \bar{q}_j - \sum_j q_j^{(i)}}{\sigma_D + \zeta} = \frac{\bar{D} - D^{(i)}}{\sigma_D + \zeta} = A^{(i)}.$$

Property (ii): Anchored signed-gradient preservation. For $a_j^{(i)} < 0$, the negative-branch loss is

$$\ell_j^{(i)} = \|v_{\theta,j}^{(i)} - v_{\mathrm{neg},j}^{(i)}\|_2^2,$$

which is non-negative and is minimized at $v_{\theta,j}^{(i)} = v_{\mathrm{neg},j}^{(i)}$. We treat $a_j^{(i)}$, $v_{\mathrm{old},j}^{(i)}$, and $v_j^{\star,(i)}$ as constants when differentiating with respect to $v_{\theta,j}^{(i)}$. At the sampling-model anchor,

$$\begin{aligned} \nabla_{v_{\theta,j}^{(i)}} \ell_j^{(i)} \Big|_{v_{\theta,j}^{(i)} = v_{\mathrm{old},j}^{(i)}} &= 2\left(v_{\mathrm{old},j}^{(i)} - v_{\mathrm{neg},j}^{(i)}\right) \\ &= 2a_j^{(i)}\left(v_{\mathrm{old},j}^{(i)} - v_j^{\star,(i)}\right), \end{aligned}$$

where the second equality follows from Eq. 13. For $a_j^{(i)} \geq 0$, differentiating the positive branch directly gives the same expression. Hence, for either branch, the local gradient at the anchor equals that of the formal signed flow-matching objective.

D Detailed Version of Algorithm 1

For notation in Alg. 2, i indexes the K candidates generated under the same condition during sample collection. After the collected samples are pooled and partitioned into mini-batches, b indexes an individual sample in the current mini-batch $\mathcal{B}_m$; hence its token count $N^{(b)} = |\mathcal{S}^{(b)}|$ may vary across samples.

E Limitations

TopoFT substantially reduces non-manifold edges, non-manifold vertices, and boundary edges when regular surface topology is desired, but does not guarantee fully manifold or watertight outputs. Residual irregularities may therefore remain in decoded meshes, although geometric fidelity is largely preserved.

For applications requiring strict manifoldness or watertightness, TopoFT can be combined with downstream mesh repair, hole filling, or remeshing to enforce additional application-specific constraints.

Algorithm 2 Detailed Topology-Aware Online Post-Training (TopoFT)

Require: Pretrained Stage-2 parameters θ_{ref}; frozen Stage-1 model; fixed Stage-3 decoding and meshing pipeline $\mathrm{Dec}_{\mathrm{mesh}}$; conditioning dataset $\mathcal{C}$; cached reference pairs $(\mathcal{S}, \mathcal{M}^{\mathrm{ref}})$ for each condition; group size K; learning rate η; stabilization constant $\zeta > 0$

$\theta \leftarrow \theta_{\mathrm{ref}}$
for each online round **do**
  $\theta_{\mathrm{old}} \leftarrow \theta, \quad \mathcal{D} \leftarrow \varnothing$
  for each sampled condition $c \sim \mathcal{C}$ **do** ▷ Topology-guided sample collection
    Retrieve cached $(\mathcal{S}, \mathcal{M}^{\mathrm{ref}})$ for condition c
    $N \leftarrow |\mathcal{S}|$
    Sample $x_0^{(i)} \sim p_{\theta_{\mathrm{old}}}(\cdot \mid c, \mathcal{S}), \quad i = 1, \dots, K$
    $\mathcal{M}^{(i)} \leftarrow \mathrm{Dec}_{\mathrm{mesh}}(x_0^{(i)}), \quad i = 1, \dots, K$
    Compute $D^{(i)} \leftarrow D(x_0^{(i)}), \quad i = 1, \dots, K$, using Eq. 5 and $\mathcal{M}^{\mathrm{ref}}$
    $\bar{D} \leftarrow K^{-1} \sum_{i=1}^{K} D^{(i)}, \quad \sigma_D \leftarrow \sqrt{K^{-1} \sum_{i=1}^{K} (D^{(i)} - \bar{D})^2}$
    Compute token-wise cost contributions $\{q_j^{(i)}\}_{j=1}^{N}, \quad i = 1, \dots, K$, using Eq. 9
    $\bar{q}_j \leftarrow K^{-1} \sum_{i=1}^{K} q_j^{(i)}, \quad j = 1, \dots, N$
    $a_j^{(i)} \leftarrow \dfrac{\bar{q}_j - q_j^{(i)}}{\sigma_D + \zeta}, \quad i = 1, \dots, K, \ j = 1, \dots, N$ ▷ Global-advantage-preserving attribution
    $\mathbf{a}^{(i)} \leftarrow \{a_j^{(i)}\}_{j=1}^{N}, \quad i = 1, \dots, K$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{ \left(c, \mathcal{S}, x_0^{(i)}, \mathbf{a}^{(i)} \right) \right\}_{i=1}^{K}$
  end for
  Partition $\mathcal{D}$ into mini-batches $\{\mathcal{B}_m\}_{m=1}^{M}$
  for $m = 1, \dots, M$ **do** ▷ Anchored dual-branch flow optimization
    Let $\mathcal{B}_m = \left\{ \left(c^{(b)}, \mathcal{S}^{(b)}, x_0^{(b)}, \mathbf{a}^{(b)} \right) \right\}_{b=1}^{|\mathcal{B}_m|}$
    for $b = 1, \dots, |\mathcal{B}_m|$ **do**
      $N^{(b)} \leftarrow |\mathcal{S}^{(b)}|, \quad \mathbf{a}^{(b)} = \{a_j^{(b)}\}_{j=1}^{N^{(b)}}$
      Sample $z^{(b)} \sim \mathcal{N}(0, 1), \quad t^{(b)} \leftarrow \mathrm{sigmoid}(1 + z^{(b)}), \quad \epsilon^{(b)} \sim \mathcal{N}(0, I)$
      $x_t^{(b)} \leftarrow (1 - t^{(b)}) x_0^{(b)} + t^{(b)} \epsilon^{(b)}$
      $v^{\star,(b)} \leftarrow \epsilon^{(b)} - x_0^{(b)}$
      $v_\theta^{(b)} \leftarrow v_\theta \left(x_t^{(b)}, t^{(b)}, c^{(b)}, \mathcal{S}^{(b)} \right)$
      $v_{\mathrm{old}}^{(b)} \leftarrow \mathrm{sg}\left[v_{\theta_{\mathrm{old}}} \left(x_t^{(b)}, t^{(b)}, c^{(b)}, \mathcal{S}^{(b)} \right) \right]$
      for $j = 1, \dots, N^{(b)}$ **do**
        if $a_j^{(b)} \geq 0$ **then**
          $\ell_j^{(b)} \leftarrow a_j^{(b)} \left\| v_{\theta,j}^{(b)} - v_j^{\star,(b)} \right\|_2^2$
        else
          $v_{\mathrm{neg},j}^{(b)} \leftarrow v_{\mathrm{old},j}^{(b)} + a_j^{(b)} \left(v_j^{\star,(b)} - v_{\mathrm{old},j}^{(b)} \right)$
          $\ell_j^{(b)} \leftarrow \left\| v_{\theta,j}^{(b)} - v_{\mathrm{neg},j}^{(b)} \right\|_2^2$
        end if
      end for
      $\ell^{(b)} \leftarrow \dfrac{1}{N^{(b)}} \sum_{j=1}^{N^{(b)}} \ell_j^{(b)}$
    end for
    $\mathcal{L}_{\mathcal{B}_m} \leftarrow \dfrac{1}{|\mathcal{B}_m|} \sum_{b=1}^{|\mathcal{B}_m|} \ell^{(b)}$
    $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\mathcal{B}_m}$
  end for
end for
Output: Fine-tuned Stage-2 model θ