

Channel Match: Calibrating Shape Latents for Few-Step 3D Generation

Di Yang¹ Zhihao Li² Yanhai Xiong¹ Yufei Wang²

¹College of William & Mary ²SparcAI Inc.

dyang06@wm.edu zhihao.li@sparclab.ai
yxiong05@wm.edu yufei.wang@sparclab.ai

Abstract

Recent native 3D generators achieve high-quality asset creation by modeling structured 3D latents with flow-based transformers, but their high-resolution generation remains computationally expensive due to costly iterative sampling. In this work, we introduce a few-step distillation framework for native 3D generation, enabled by continuous self-consistency training with a custom JVP FlashAttention kernel and an analytic velocity regression objective for sparse structured 3D latents. To stabilize joint optimization, we further introduce an adaptive loss weighting strategy that automatically balances them without manual tuning. Beyond acceleration, we reveal a previously overlooked failure mode of few-step native 3D generation: sampling-induced latent distribution drift preserves geometric correctness but disrupts downstream PBR texture synthesis, causing texture artifacts on geometrically valid assets. To address this issue, we propose Channel Match, a training-free adaptive per-channel latent calibration method that restores the latent distribution without modifying either generator. We validate our approach on TRELLIS.2 at the 1024³ resolution setting. Our two-step model achieves 4.5× faster geometry sampling while maintaining geometry quality comparable to the many-step teacher, and Channel Match further improves rendering quality by recovering the latent interface between geometry and appearance generation.

1 Introduction

Recent advances in native 3D generators have enabled high-quality asset creation by directly modeling structured 3D representations. These models encode assets into sparse and structured latent spaces and generate them with flow-based transformers [Xiang et al., 2025, Liu et al., 2023, Lipman et al., 2022]. Despite their impressive details, high-resolution native 3D generators remain computationally expensive, as each flow step requires processing a large number of 3D tokens in structured latent spaces, and generation typically requires many iterative sampling steps. This computational cost limits their deployment in interactive applications.

Few-step distillation provides a promising direction to re-

duce this cost. Consistency models and their continuous-time formulations compress multi-step samplers into one or two steps [Song et al., 2023, Lu and Song, 2025], while distribution-matching and adversarial distillation methods achieve near-teacher quality with even fewer evaluations [Yin et al., 2024b, Lin et al., 2024, Sauer et al., 2024]. However, directly applying these techniques to native 3D generation remains challenging due to fundamental differences in latent representations and generation pipelines.

First, native 3D generators rely on sparse and variable-length structured latents, whereas existing continuous-time distillation methods are primarily designed for dense fixed-dimensional representations [Song et al., 2023, Lu and Song, 2025]. Such latent structures are essential for scaling 3D generation to high resolutions, but introduce challenges for computing consistency objectives, particularly the velocity directional derivatives required by continuous-time formulations. Second, native 3D generation pipelines contain intermediate latent interfaces shared across multiple generation stages. Unlike image generation where the latent is directly decoded into pixels, a geometry latent in native 3D generation can serve as the input to downstream appearance synthesis, such as PBR texture generation. Therefore, reducing sampling steps may preserve geometric structure while shifting latent statistics away from the distribution expected by downstream generators, causing quality degradation that is invisible to geometry-only evaluation.

In this work, we investigate few-step distillation for high-resolution native 3D generation. We develop a distillation framework tailored for sparse structured 3D latents, enabling continuous self-consistency training through a custom JVP FlashAttention kernel. To avoid the ambiguity of learning a consistent but incorrect transport field, we incorporate an analytic velocity regression objective that anchors the student toward the true marginal transport. Furthermore, we introduce an adaptive loss weighting strategy that automatically balances self-consistency and velocity regression objectives across timesteps without manual tuning of loss weights.

Beyond acceleration, we reveal a previously overlooked failure mode of few-step 3D generation: sampling-induced latent distribution drift. The accelerated geometry sampler can produce structurally valid shape latents while deviating from the



Figure 1: The few-step distribution shift and its correction by **CM**. From left: input image; the untextured mesh and PBR render of our two-step student without Channel Match, where the sampling-induced shifted shape latent drives the frozen PBR texture model to produce faulty surface materials (dark/blotchy patches); and the mesh and render with **CM**. For both few-step teacher and student sampling, calibrating the latent removes the texture artifacts, while the geometry remains unchanged. (We consider samples with simple geometry here, where both the two-step teacher and our two-step student already recover structures close to the many-step reference. This controlled setting isolates the effect of latent distribution shift rather than geometry reconstruction errors, showing that the shift is a general issue of few-step sampling and can be effectively corrected by **CM**.)

training distribution in channel statistics. When consumed by the downstream PBR texture generation stage, these shifted latents lead to incorrect material synthesis, producing surface artifacts on otherwise geometrically valid assets. To address this issue, we propose Adaptive Channel Match, a training-free adaptive per-channel latent calibration method that restores sampling-shifted latents toward the training distribution without modifying either generator.

We validate our approach on TRELLIS.2 [Xiang et al., 2026] at the 1024^3 resolution setting. Our two-step model achieves $4.5\times$ faster geometry sampling while maintaining geometry quality comparable to the many-step teacher. Adaptive channel match further improves rendering quality by recovering the latent interface between geometry generation and appearance synthesis.

Our contributions are summarized as follows.

- We reveal a previously overlooked failure mode in few-step native 3D generation: aggressive sampling reduction can introduce distribution shifts in the geometry latent that preserve geometric completeness but violate the latent interface with downstream appearance synthesis, causing texture artifacts that resemble structural defects.
- We propose Channel Match, a training-free adaptive per-channel latent calibration method that restores the shifted geometry latent toward the training distribution. Without retraining either the geometry or texture generator, it eliminates texture artifacts, improves paired rendering quality toward the many-step teacher, and mitigates similar degradation in naively few-step sampling.
- We develop a scalable few-step distillation framework for high-resolution native 3D flow models. By enabling continuous self-consistency training on sparse structured 3D latents with a custom JVP FlashAttention kernel and an

analytic velocity regression objective, our two-step student achieves $4.5\times$ faster geometry sampling with quality comparable to the many-step teacher and surpassing SOTA baselines.

2 Related Work

Native 3D generation. Structured-latent models generate features on sparse active voxels via rectified-flow transformers, decoding to meshes, Gaussians, or radiance fields [Xiang et al., 2025, Ren et al., 2024, Kerbl et al., 2023]. TRELLIS.2 extends this line to 1024^3 -grid latents with PBR materials and a separate texture stage [Xiang et al., 2026]. High-resolution sparse representations similarly enable detailed geometry at large grid sizes [Li et al., 2026, He et al., 2025, Zhang et al., 2024]. Other native generators diffuse compact vector-set or primitive latents [Lai et al., 2025, Li et al., 2025, Chen et al., 2025c, Zhao et al., 2025, Chen et al., 2025a], and earlier feed-forward or latent-field methods trade fidelity for speed [Tang et al., 2024, Lan et al., 2025]. Supported by large datasets [Deitke et al., 2023b,a], we fix the shape and texture stages of TRELLIS.2 and distill only the shape flow, so any measured change stems from the shape latent alone.

Few-step distillation. Progressive distillation halves the step count repeatedly [Salimans and Ho, 2022]; consistency models regress points on a trajectory to a shared endpoint [Song et al., 2023], and the continuous-time form removes the discretization schedule once its training instability is controlled through a simplified parameterization and an adaptive weighting [Lu and Song, 2025, Karras et al., 2022]. A second family replaces trajectory matching with a distribution-level objective: distribution-matching, adversarial, and unified trajectory-distribution distillations [Yin et al., 2024b,a, Sauer et al., 2024, Lin et al., 2024, Luo et al., 2025],

which unifies the two, reach one-to-few-step image generation. These methods rely on 2D image discriminators, whose signal is hard to obtain for a native 3D latent because quality evaluation requires non-differentiable rendering. We therefore use continuous-time self-consistency in velocity space, paired with a ground-truth velocity loss to anchor the fixed point.

Forward-mode attention and 3D consistency. The continuous objective needs a forward-mode tangent through attention. FlashAttention [Dao et al., 2022] and its sparse extension SCFA [Pagliardini et al., 2023] keep attention memory-efficient but expose only the primal output. The closest 3D consistency model, ConTiCoM-3D [Eilermann et al., 2026], avoids Jacobian-vector products entirely because they are expensive for unordered point clouds, and works teacher-free in raw point space. Ours is the opposite: a teacher-based, latent-space, sparse-voxel pipeline where a custom fused kernel makes the Jacobian-vector product tractable at high resolution.

Accelerating 3D generation. Direct transfer of 2D distillation to native 3D is uncommon. FlashVDM accelerates both the sampler and the decoder of a vector-set model but generates shape without appearance [Lai et al., 2025]. A parallel line instead cuts attention cost with sparse attention [Chen et al., 2025b, Wu et al., 2025]. The marginal-data-transport distillation of Zhou et al. [2026] converts the transport objective into velocity matching and velocity distillation and reaches one-to-two-step generation on an earlier structured-latent backbone; it is the closest baseline, and we reimplement it on TRELLIS.2 for a controlled comparison. Neither line reports the latent distribution-shift failure that Channel Match targets, which appears once the shape stage is pushed to very few steps.

Statistics alignment. Matching per-channel mean and variance to a reference is a long-standing idea, from batch normalization [Ioffe and Szegedy, 2015] to adaptive instance normalization for style transfer [Huang and Belongie, 2017]. Channel Match uses the same primitive but contributes a diagnosis, not a layer: a few-step shape latent drifts off the training marginals, and realigning its per-channel statistics restores a frozen downstream stage.

3 Preliminaries

3.1 Structured-Latent Generation in TRELLIS.2

An asset is encoded as a sparse structured latent $x = \{(x_i, p_i)\}_{i=1}^L$, where p_i indexes an active voxel of a high-resolution grid and $x_i \in \mathbb{R}^C$ is its feature. Generation factorizes into three conditional flow stages driven by the input image c : a sparse-structure stage predicts the active voxels; a shape stage generates the geometry latent on those voxels; and a texture stage generates the PBR material latent. A decoder $\mathcal{D}$ maps the shape and texture latents to a textured mesh. The

shape stage is coarse-to-fine: a low-resolution (512) flow produces an initial latent whose active coordinates are upsampled to the 1024 grid, and a high-resolution (1024) flow generates the final geometry latent on those coordinates. We distill this 1024 shape flow. The sparse-structure stage, the 512 flow that seeds the coordinates, and the texture stage are frozen and shared across all configurations leaving only the 1024 shape flow to differ between teacher and student.

3.2 Rectified Flow

Let x_0 be a data latent and $\epsilon \sim \mathcal{N}(0, I)$ noise. Rectified flow interpolates along a straight path

$$x_t = (1 - t)x_0 + (\sigma_{\min} + (1 - \sigma_{\min})t)\epsilon, \quad t \in [0, 1], \quad (1)$$

with a small $\sigma_{\min}$, so $t = 0$ is data and $t = 1$ is noise. The instantaneous velocity of this path is

$$u(x_t, t) = \frac{dx_t}{dt} = (1 - \sigma_{\min})\epsilon - x_0. \quad (2)$$

A flow model v_ϕ is trained to regress the conditional expectation $\bar{v}(x_t, t) = \mathbb{E}[u | x_t]$, which defines the probability-flow ODE $\dot{x}_t = \bar{v}(x_t, t)$. Samples are drawn by integrating this ODE from $t = 1$ to $t = 0$; the teacher uses a K_ϕ -step Euler schedule, and our goal is a student v_θ whose K -step Euler integral, with $K \in \{1, 2, 4\}$, matches the teacher’s output.

3.3 Continuous-Time Self-Consistency

The consistency function $f(x_t, t)$ maps any point on a probability-flow trajectory to that trajectory’s data endpoint [Song et al., 2023]. Consistency holds when f is constant along the trajectory, i.e. its total time derivative vanishes, $\frac{d}{dt}f(x_t, t) = 0$. Expressed through the velocity, this is a differential constraint that a single-step map must satisfy at every t . The continuous-time formulation of Lu and Song [2025] makes this constraint trainable by a simplified parameterization and an adaptive per-timestep weighting; we adopt it in velocity space, which matches the rectified-flow backbone, and combine it with Equation 2.

4 Method

We distill the many-step teacher v_ϕ into a few-step student v_θ , initialized from the teacher, with two velocity-space losses that constrain different properties of the field and a per-timestep weighting that balances them.

4.1 Ground-Truth Velocity Regression

The first loss regresses the student velocity onto the analytic target of Equation 2,

$$\mathcal{L}_{\text{gt}} = \mathbb{E}_{x_0, \epsilon, t} \left[\left\| v_\theta(x_t, t) - ((1 - \sigma_{\min})\epsilon - x_0) \right\|_2^2 \right]. \quad (3)$$

Its minimizer is the true marginal field, so it fixes the values the student should take, and expressing it in velocity space

avoids the small- t weighting pathology of a sample-space reconstruction. This loss alone does not yield a few-step sampler; it anchors the next loss away from a consistent but incorrect transport.

4.2 Continuous Self-Consistency

The second loss makes the field invariant to the number of integration steps. Along a teacher trajectory, the total time derivative of the student velocity is

$$g(x_t, t) = \frac{d}{dt} v_\theta(x_t, t) = \partial_t v_\theta + (\nabla_x v_\theta) \bar{v}_\phi(x_t, t), \quad (4)$$

where the transport direction is the teacher velocity $\bar{v}_\phi$. We evaluate Equation 4 with one forward-mode (JVP) pass: the dual tangent of the latent is seeded with $\bar{v}_\phi$, giving the spatial term $(\nabla_x v_\theta) \bar{v}_\phi$, and the dual tangent of the time input is seeded so that the pass also returns $\partial_t v_\theta$. Concretely, the backbone consumes a scaled time $1000t$, and seeding its tangent with the constant 1000 yields $\partial_t v_\theta$ in the same pass. Omitting the time term, or seeding the spatial term with anything other than the teacher velocity, biases the tangent.

We normalize the per-token tangent with a smooth, magnitude-preserving robust cap. With $r = \|g\|/(\kappa \cdot \text{med}\|g\|)$ per token and a clip factor κ ,

$$\hat{g} = \frac{\tanh(r)}{r} g. \quad (5)$$

As $r \rightarrow 0$ the scale tends to 1 (small tangents preserved, correction vanishing once $g \rightarrow 0$); as $r \rightarrow \infty$ the token norm saturates near $\kappa \cdot \text{med}\|g\|$. Unlike unit-norm normalization, this keeps small consistency signals intact. The self-consistency target regresses the student toward a stop-gradient, exponential-moving-average (EMA) copy $\tilde{v}_\theta$ of itself, moved along the normalized tangent,

$$v^* = \text{sg}(\tilde{v}_\theta(x_t, t) - \eta \omega t \hat{g}), \quad (6)$$

$$\mathcal{L}_{\text{scm}} = \mathbb{E}[\|v_\theta(x_t, t) - v^*\|_2^2], \quad (7)$$

where η is a step scale and ω a warm-up factor ramped from 0 to 1 over the first training steps. Anchoring the value to the EMA student rather than the teacher avoids a biased fixed point that arises when a slightly mismatched teacher value is used as both the target and the transport direction.

4.3 Adaptive Loss Balancing

The two losses differ in scale and vary with t . Rather than tune a per- t schedule, we apply an uncertainty weighting: for a per-sample loss ℓ and a learnable scalar $w(t)$, the weighted term is $\ell e^{-w(t)} + w(t)$, whose minimizer $w^*(t) = \log \ell$ self-normalizes the term across time [Karras et al., 2022, Lu and Song, 2025]. We realize $w(\cdot)$ as a small Fourier-feature multilayer perceptron trained jointly, and apply it to both terms, giving

$$\mathcal{L} = \mathbb{E}[e^{-w_{\text{scm}}(t)} \ell_{\text{scm}} + w_{\text{scm}}(t) + e^{-w_{\text{gt}}(t)} \ell_{\text{gt}} + w_{\text{gt}}(t)]. \quad (8)$$

The reported models use the two velocity-space losses of Equation 8.

4.4 JVP FlashAttention for Sparse Latents

Equation 4 is a Jacobian-vector product, so evaluating it requires forward-mode differentiation through every layer of the transformer, including attention. Stock FlashAttention [Dao et al., 2022] returns the output but not its forward-mode tangent, and would forfeit its memory savings on the high-resolution sparse latent. We implement a custom JVP FlashAttention kernel that, given the primal and tangent of the packed query-key-value tensor of the variable-length sparse latent, returns the primal output and its tangent in a single fused pass, and we patch the variable-length attention entry points used by the backbone to call it during the tangent computation. The kernel keeps the tangent evaluation near the memory cost of standard attention (Section 5.8), making the continuous objective usable at high resolution. We give the full kernel implementation in the supplementary material. Algorithm 1 summarizes one training step.

Algorithm 1: Few-step shape distillation and generation

Training (one step). **Input:** teacher v_ϕ , student v_θ , EMA student $\tilde{v}_\theta$, weighting net w ; clip κ , step η , warm-up ω

- 1: Draw a token-balanced micro-batch; load latents x_0 , conditions c
- 2: $\epsilon \sim \mathcal{N}(0, I)$, $t \sim \mathcal{U}(0, 1)$; form x_t by Eq. 1
- 3: $\bar{v} \leftarrow v_\phi(x_t, t, c)$ // teacher direction, no grad
- 4: $g \leftarrow \partial_t v_\theta + (\nabla_x v_\theta) \bar{v}$ via JVP (dual seeds $\bar{v}$, 1000)
- 5: $\hat{g} \leftarrow (\tanh r / r) g$, $r = \|g\|/(\kappa \text{med}\|g\|)$
- 6: $\tilde{v} \leftarrow \tilde{v}_\theta(x_t, t, c)$; $v \leftarrow v_\theta(x_t, t, c)$
- 7: $v^* \leftarrow \text{sg}(\tilde{v} - \eta \omega t \hat{g})$; $v_{\text{gt}} \leftarrow (1 - \sigma_{\min})\epsilon - x_0$
- 8: $\ell_{\text{scm}} \leftarrow \|v - v^*\|_2^2$; $\ell_{\text{gt}} \leftarrow \|v - v_{\text{gt}}\|_2^2$
- 9: $\mathcal{L} \leftarrow$ uncertainty-weighted sum (Eq. 8); **backprop**; update v_θ , w ; EMA update $\tilde{v}_\theta$

Generation with Channel Match. **Input:** image c ; frozen prefix, texture flow, decoder $\mathcal{D}$; distilled v_θ ; stats $(\hat{\mu}, \hat{\sigma})$; threshold τ_S ; steps K

- 1: $S \leftarrow$ sparse structure(c); upsample to 1024 coords // shared prefix
 - 2: $x \leftarrow K$ -step Euler integral of v_θ from noise on those coords
 - 3: $S \leftarrow \frac{1}{C} \sum_c \max(0, \hat{\sigma}_c / \sigma_c^s(x) - 1)$ // Eq. 10
 - 4: **if** $S > \tau_S$ **then**
 - 5: $x \leftarrow$ ChannelMatch(x ; $\hat{\mu}, \hat{\sigma}$) // Eq. 9
 - 6: **end if**
 - 7: $y \leftarrow$ texture flow(x, c); $M \leftarrow \mathcal{D}(x, y)$; **return** M
-

4.5 Channel Match

After distillation the decoded mesh is geometrically complete, but the generated shape latent moves off the training distribution in its per-channel value statistics, and the frozen PBR texture stage, trained only on in-distribution latents, produces incorrect materials in the affected regions (Figure 1). The geometry is intact, only the appearance degrades. Channel

Match calibrates the shape latent to remove this shift before it reaches the texture stage.

For a latent $x = \{(x_i, p_i)\}_{i=1}^L$ with $x_i \in \mathbb{R}^C$, we treat each channel c independently. The stored training statistics $(\hat{\mu}_c, \hat{\sigma}_c)$ are the mean and standard deviation of channel c over all active tokens of the training set, computed once offline; the sampled statistics (μ_c^s, σ_c^s) are the same two quantities computed over the active tokens of the current generated latent. With a strength s , the calibrated feature is

$$x'_c = \hat{\mu}_c + (x_c - \mu_c^s) \left(1 + s(\hat{\sigma}_c / \sigma_c^s - 1)\right). \quad (9)$$

Only dataset-level statistics enter, so the calibration is training-free and modifies neither model. It is a per-channel affine map and does not model inter-channel correlation, which we revisit in the limitations. We evaluate two forms, full ($s=1$, matching mean and std) and std-only (keeping the asset’s mean); a mean-only variant is not considered, since the observed drift is a collapse of the per-channel standard deviation (Table 6).

The shift depends on the step count, so we gate the calibration by a collapse score computed from the sampled latent alone,

$$S = \frac{1}{C} \sum_c \max\left(0, \frac{\hat{\sigma}_c}{\sigma_c^s} - 1\right), \quad (10)$$

and apply it only when S exceeds a threshold τ_S , so no ground truth is needed at inference. Algorithm 1 places Channel Match as a single in-place step between the distilled shape flow and the frozen texture flow; the shared prefix and the texture stage are unchanged.

5 Experiments

5.1 Implementation Details

The student is initialized from the pretrained TRELLIS.2-4B shape model, and we distill only the 1024-resolution shape flow; the sparse-structure stage, the 512 shape flow, and the texture stage are frozen. Training runs on eight A800-80GB GPUs with distributed data parallelism. All experiments use a fixed random seed of 42 for reproducibility. Because sparse latents have a variable number of active voxels, we don’t use a fixed batch size; instead we pack assets into each step until a token budget of 40,000 active tokens is reached and accumulate gradients, balancing per-step compute across differently sized assets. We use Adam with learning rate 2×10^{-6} , $(\beta_1, \beta_2) = (0.0, 0.999)$, and gradient clipping at 1.0; the weighting network uses learning rate 10^{-2} . The EMA decay is 0.9999, $\sigma_{\min} = 10^{-5}$, the tangent clip is $\kappa = 4$ times the running median, the tangent warm-up is 1000 steps, and the step scale is $\eta = 1$; the two losses are equally weighted and training runs up to 80,000 steps in bfloat16. The JVP FlashAttention kernel is a fused Triton kernel that patches the backbone’s variable-length attention during the tangent pass. The channel statistics $(\hat{\mu}, \hat{\sigma})$ for Channel Match are estimated once on the training latents.

We compare our student at one, two, and four steps against the pretrained teacher run at twelve steps (a many-step upper bound) and at two steps (a naive few-step baseline), the marginal-data-transport distillation of Zhou et al. [2026] reimplemented on this backbone (MDT), and a self-consistency-only ablation of our method that drops the ground-truth loss $\mathcal{L}_{\text{gt}}$ (SCM). All configurations share the same frozen prefix and texture stage.

5.2 Datasets

We distill on a 100K-asset subset of Objaverse and Objaverse-XL [Deitke et al., 2023b,a], each asset providing a ground-truth shape latent and a rendered conditioning image. For evaluation we use all 3,216 assets of the Toys4K object set [Stojanov et al., 2021], each conditioned on a single rendered view.

5.3 Metrics

Geometry and appearance are scored under a fixed textured-mesh protocol: four views at 30° elevation and azimuths $\{0, 90, 180, 270\}$. We report set-level Fréchet distance in Inception [Szegedy et al., 2016, Heusel et al., 2017] and DINOv2 [Oquab et al., 2024, Stein et al., 2023] feature spaces over the renders and ULIP-2 image-shape cosine similarity [Xue et al., 2024] averaged over views. We render appearance from the PBR-textured mesh that TRELLIS.2 produces. The starred baselines in Table 1 use varying representations (Gaussians, neural fields, primitive sets) and evaluation protocols. They serve only as loose references. We draw controlled conclusions strictly from the lower block, where all configurations share the identical frozen prefix, texture stage, and render protocol. CM acts locally, so a set-level Fréchet distance is the wrong instrument for it; we instead take the many-step teacher’s render as the reference and report paired LPIPS [Zhang et al., 2018], SSIM [Wang et al., 2004], and PSNR between each student render and the teacher render under the same protocol, with the fraction of assets each calibration moves toward the teacher. Timing is measured on a single A800-80GB GPU.

5.4 Distillation: Geometry and Efficiency

Table 1 reports the main comparison. Among two-step configurations, our student attains the best Inception and DINOv2 Fréchet distances and the highest ULIP, improving on the two-step teacher, on the reimplemented marginal-data-transport baseline, and on the self-consistency-only variant, and it matches the twelve-step teacher on ULIP. Figure 2 shows the same ordering qualitatively, where MDT fractures the geometry and SCM produces less complete geometry than ours. The four-step student narrows most of the remaining Fréchet-distance gap. Table 2 reports timing: distilling the shape sampler to two steps cuts shape sampling from 5.29 s to 1.18 s per asset (4.5 $\times$) and end-to-end image-to-shape generation from 11.51 s to 7.55 s per asset (1.5 $\times$); the end-to-end factor

is bounded by the shared prefix and the unchanged texture stage.

5.5 Channel Match Improves the Texture

Table 3 reports the paired render comparison on the demonstration assets. While the uncalibrated two-step student tracks the teacher, both Channel Match forms improve all metrics. The full match optimizes LPIPS (0.0802 to 0.0747), SSIM (0.8819 to 0.8884), and PSNR (24.44 to 24.89), moving the student closer to the teacher on 73%–88% of assets.

5.6 Ablations

Form of the calibration. Table 4 isolates the calibration form on the two-step student over the full Toys4K set. On these geometrically simple assets, where the artifact is rare, matching only the standard deviation is the safe choice: it slightly improves the Inception Fréchet distance, whereas the full match, which also moves the mean, is a touch worse than doing nothing. This is the opposite of the demonstration assets in Table 3, where the artifact is common and the full match helps most. The two observations agree: the full match helps most where the collapse is severe (the artifact-heavy demonstration assets), and the std-only match is the safer choice where it is mild (the simple Toys4K assets). The same std match lowers both Fréchet distances for the naively two-step-sampled teacher (Table 4, lower block), so the shift is distributional rather than specific to the distilled student.

Adaptive gate. As an upper bound, an oracle that keeps, per asset, whichever of the uncalibrated or calibrated latent is closer to the ground truth lowers the two-step Inception Fréchet distance of Table 1 from 14.09 to 13.53 while calibrating about half of the assets, so per-asset gating has headroom. The deployable gate of Equation 10 approximates this without ground truth, applying the calibration only when the collapse score exceeds a threshold τ_S . Table 5 sweeps τ_S on the demonstration assets: LPIPS, SSIM, and PSNR are essentially flat once the match is applied to at least 75% of assets, so the result is insensitive to τ_S in this range. The default $\tau_S=0.20$ gives the best LPIPS and PSNR, and always-on calibration is a safe fallback. The oracle bound above is on Toys4K FD, so it is not directly comparable to this sweep; closing the gap to it is left open.

Losses. Removing the ground-truth velocity loss and keeping only the self-consistency loss (the SCM row of Table 1) raises the two-step Inception Fréchet distance from 14.09 to 14.76 and the DINOv2 Fréchet distance from 127.44 to 135.02, and lowers ULIP, confirming that the two losses are complementary: self-consistency makes the field step-agnostic while the velocity loss anchors its values. Removing instead the self-consistency loss leaves a correct but not step-agnostic field that degrades sharply at two steps, the standard failure of few-step flow sampling without distillation.

5.7 Latent Drift and Sampling Steps

Table 6 quantifies the shift of Section 4.5 over Toys4K. The latent is per-channel standardized, so $\hat{\sigma}_c = 1$ and the drift is one-sided: sampled channels fall below the training std. The collapse score S (Eq. 10) sums exactly this deficit over channels, so it is a direct scalar measure of the shift. Both S and the fraction of channels with sampled std below 0.8 grow monotonically as steps shrink, and the four-step student matches the teacher’s level, consistent with its being the strongest few-step configuration in Table 1. The intervention (Table 3) shows the shift causes the artifact: matching the statistics removes it, and the monotonic trend here is consistent with the same account.

5.8 JVP Kernel Cost and Stability

Table 7 profiles the fused kernel on an A800-80GB GPU. The forward-mode tangent adds training-time cost only: inference remains unaffected. The essential property is memory rather than latency. Against a naive `torch.func.jvp` baseline, our kernel is 10.9× faster (41.7 vs 454.4,ms) and $O(N)$ rather than $O(N^2)$ in memory, so where the naive path needs 67.6,GB and runs out of memory at the $N=30,000$ production length, our kernel still runs. This $O(N)$ memory is what lets the continuous objective fit on the high-resolution sparse latent. The tangent is numerically stable: over 58,952 tokens we saw no NaN or Inf, and the cap of Equation 5 at $\kappa = 4$ roughly halves the maximum tangent norm (175.0 to 82.3) while leaving the median near 21, so it stabilizes the objective without distorting typical tangents.

6 Limitations

We distill only the shape stage, so the end-to-end speedup is bounded by the frozen texture sampler, and distilling it is a direct next step. Channel Match corrects the per-channel distribution shift and removes the resulting texture artifacts, but it does not synthesize new geometric detail, so the four-step student stays strongest on geometry-sensitive metrics; being a per-channel affine map, it also ignores inter-channel and spatial structure, which a full-covariance or grouped variant could capture. The collapse-score gate already recovers most of the available gain, and narrowing the remaining margin to the ground-truth oracle is a promising direction. Extending the same calibration to the texture stage is a natural next question we leave open.

7 Conclusion

We distilled the high-resolution shape stage of TRELLIS.2 into a two-step student, using a continuous self-consistency loss, made tractable on the sparse high-resolution latent by a custom JVP FlashAttention kernel, together with an analytic ground-truth velocity loss. At two steps the student reaches geometry comparable to the many-step teacher, matching it on ULIP with a small residual Fréchet-distance gap, at 4.5× faster

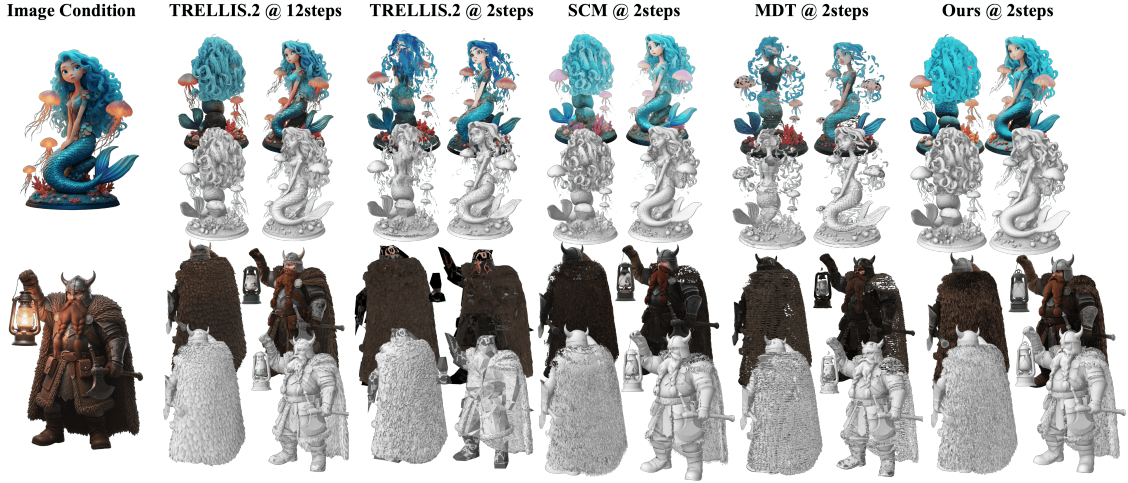


Figure 2: Qualitative comparison on TRELIS.2 demonstration assets. From left: input image, the 12 and 2-step teacher, SCM @2, MDT @2, and ours @2; each block shows the PBR render and untextured mesh. MDT fractures the geometry, SCM leaves surface artifacts, and ours is closest to the teacher.

Method	Steps	Time (s)	FD-Incep ↓	FD-DINOv2 ↓	ULIP-I ↑
<i>Prior methods (cited; Toys4K subset, each method's own protocol)</i>					
LGM* [Tang et al., 2024]	–	5	26.31	322.71	–
3DTopia-XL* [Chen et al., 2025c]	25	5	37.68	437.37	–
Ln3Diff* [Lan et al., 2025]	250	8	26.61	357.93	–
TRELIS* [Xiang et al., 2025]	25×2	–	9.35	67.21	–
<i>Our pipeline (TRELIS.2 backbone; full Toys4K; fixed textured-mesh protocol)</i>					
Teacher @12 (upper bound)	12	11.51	12.56	106.66	38.45
Teacher @2	2	–	15.67	155.40	38.23
MDT [Zhou et al., 2026] @2 (reimpl.)	2	–	15.98	132.80	38.04
SCM @2 (ours, no $\mathcal{L}_{gt}$)	2	–	14.76	135.02	38.43
Ours @1	1	–	17.56	152.58	38.21
Ours @2	2	7.55	14.09	127.44	38.48
Ours @4	4	–	13.40	113.56	38.49

Table 1: Main comparison on Toys4K. ULIP-I is $\times 100$; Fréchet distances are set-level. Starred rows are cited from prior work under each method’s own protocol and are not directly comparable to the lower block. Times are A800-80GB end-to-end image-to-shape latency where measured.

Stage	Teacher	Ours @2
Shape sampling (s/asset)	5.29	1.18 (4.5×)
End-to-end image→shape (s/asset)	11.51	7.55 (1.5×)

Table 2: Latency on a single A800-80GB GPU.

shape sampling, improving on a marginal-data-transport baseline and a self-consistency-only variant. We further identified a few-step failure mode, an out-of-distribution shape latent that drives the frozen texture model to produce artifacts, and resolved it with Channel Match, a training-free per-channel calibration on the shape latent that removes the texture artifacts and moves the student’s appearance toward the teacher, and that reduces the same degradation for a naively few-step-sampled teacher, without retraining either model.

Calibration	LPIPS ↓	SSIM ↑	PSNR ↑
None (two-step student)	0.0802	0.8819	24.44
Std only	0.0762	0.8860	24.72
Full (mean + std)	0.0747	0.8885	24.85
<i>Fraction of assets moved toward teacher (↑)</i>			
Std only	73%	75%	67%
Full (mean + std)	73%	88%	73%

Table 3: Channel Match on the TRELIS.2 demonstration assets ($n=72$), scored against the twelve-step teacher’s renders under the four-view protocol. Both forms move the two-step student toward the teacher; the full match gives the larger gain on these artifact-heavy assets.

Calibration	ULIP-I $\uparrow$	FD-Incep $\downarrow$	FD-DINOv2 $\downarrow$
<i>Student @2</i>			
None	38.48	14.09	127.44
Std only	38.46	13.88	126.88
Full (mean + std)	38.38	14.16	127.88
<i>Teacher @2</i>			
None	38.23	15.67	155.40
Std only	38.28	15.21	153.12
Full (mean + std)	38.35	15.66	155.03

Table 4: Calibration form at two steps on the full Toys4K set, for both the distilled student and the naively teacher.

τ_S	applied	LPIPS $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$
$-\infty$ (always)	100%	0.0747	0.8885	24.85
0.20 (default)	77%	0.0742	0.8884	24.89
0.27	50%	0.0761	0.8862	24.72
$+\infty$ (never)	0%	0.0802	0.8819	24.44

Table 5: Sensitivity to the gating threshold τ_S on the demonstration assets ($n=72$, full mean+std match), scored by paired LPIPS/SSIM/PSNR against the twelve-step teacher under the four-view protocol.

Config	Steps	Collapse $S \downarrow$	Collapsed chans. $\downarrow$
Teacher	12	0.150	16.8%
Student	4	0.146	17.5%
Student	2	0.174	22.4%
Student	1	0.207	28.4%

Table 6: Latent drift versus sampling steps on Toys4K. Collapse score S (Eq. 10) is the mean per-channel std deficit; it and the fraction of channels with sampled std below 0.8.

Method	Lat. (ms) $\downarrow$	VRAM (GB) $\downarrow$
FlashAttn foward (no JVP) [†]	4.8	—
Naive foward	114.4	27.8
Naive foward + JVP	454.4	67.6
Triton JVP (ours)	41.7	~0.2

Table 7: Fused JVP attention vs. the naive PyTorch baseline (A800-80GB, bf16, $N=15,000$). [†]FlashAttention (forward only, no tangent) is a speed *ceiling*, not a JVP baseline.

References

- Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16251–16261, 2025a.
- Yiwen Chen, Zhihao Li, Yikai Wang, Hu Zhang, Qin Li, Chi Zhang, and Guosheng Lin. Ultra3d: Efficient and high-fidelity 3d generation with part attention. *arXiv preprint arXiv:2507.17745*, 2025b.
- Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26576–26586, 2025c.
- Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 35799–35813, 2023a.
- Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsanit, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b.
- Sebastian Eilermann, René Heesch, and Oliver Niggemann. Conticom-3d: A continuous-time consistency model for 3d point cloud generation. In *International Conference on 3D Vision*, 2026.
- Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14822–14833, 2025.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
- Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE*

- international conference on computer vision (ICCV)*, pages 1501–1510, 2017.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456. pmlr, 2015.
- Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 26565–26577, 2022.
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Fuyun Wang, Huiwen Shi, Xianghui Yang, Qingxiang Lin, Jingwei Huang, Yuhong Liu, et al. Unleashing vecset diffusion model for fast shape generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2523–2533, 2025.
- Yushi Lan, Fangzhou Hong, Shangchen Zhou, Shuai Yang, Xuyi Meng, Yongwei Chen, Zhaoyang Lyu, Bo Dai, Xingang Pan, and Chen Change Loy. Ln3diff++: Scalable latent neural fields diffusion for speedy 3d generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TAPAMI)*, 2025.
- Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TAPAMI)*, 2025.
- Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pages 118582–118600, 2026.
- Shanchuan Lin, Anran Wang, and Xiao Yang. SDXL-Lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2022.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International conference on learning representations (ICLR)*, 2023.
- Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17719–17728, 2025.
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.
- Matteo Pagliardini, Daniele Paliotta, Martin Jaggi, and François Fleuret. Faster causal attention over large sequences through sparse flash attention. *arXiv preprint arXiv:2306.01160*, 2023.
- Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4209–4219, 2024.
- Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.
- Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision (ECCV)*, pages 87–103. Springer, 2024.
- Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, pages 32211–32252, 2023.
- George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Vilecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 3732–3784, 2023.
- Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1798–1808, 2021.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.
- Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision (ECCV)*, pages 1–18. Springer, 2024.

- Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, yajie bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, and Yao Yao. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.
- Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 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 *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 14419–14429, June 2026.
- Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, et al. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27091–27101, 2024.
- Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024a.
- Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 6613–6623, 2024b.
- Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Trans. Graph.*, 43(4), July 2024. ISSN 0730-0301. doi: 10.1145/3658146.
- Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.
- Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.
- Zanwei Zhou, Taoran Yi, Jiemin Fang, Chen Yang, Lingxi Xie, Xinggang Wang, Wei Shen, and Qi Tian. Few-step flow for 3d generation via marginal-data transport distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pages 13853–13861, 2026.

A JVP FlashAttention for Sparse Latents

This appendix gives the full implementation of the fused forward-mode (JVP) attention kernel used to train the continuous self-consistency objective, and its integration with PyTorch forward-mode autodiff. We first restate why a custom kernel is needed, derive the forward-mode tangent of softmax attention, describe the fused online-softmax algorithm that evaluates it in $O(N)$ memory, and finally give the autograd bridge that exposes the kernel to `torch.autograd.forward_ad`.

A.1 Why a Custom Kernel

The continuous self-consistency loss requires the total time derivative of the student velocity along a teacher trajectory,

$$g(x_t, t) = \partial_t v_\theta + (\nabla_x v_\theta) \bar{v}_\phi, \quad (11)$$

which is a Jacobian-vector product (JVP): a single forward-mode pass through the backbone with the latent tangent seeded by the teacher velocity $\bar{v}_\phi$ and the time tangent seeded by $d(1000t)/dt = 1000$. Forward-mode differentiation must propagate a tangent through *every* layer, including attention. Stock FlashAttention [Dao et al., 2022] returns only the primal output and exposes no tangent, and a naive PyTorch attention that supports forward-mode AD (`torch.func.jvp`) materializes the $N \times N$ score matrix, whose $O(N^2)$ memory is infeasible at the 1024^3 sparse-latent token counts. We therefore implement a fused kernel that returns the primal output and its tangent in one online-softmax pass at the memory cost of standard attention.

A.2 Forward-Mode Tangent of Softmax Attention

Let $S = QK^\top/\sqrt{d}$ be the scores, $P = \text{softmax}(S)$ the row-normalized weights, and $O = PV$ the output. Given input tangents $(\dot{Q}, \dot{K}, \dot{V})$, the score tangent is

$$\dot{S} = (\dot{Q} K^\top + Q \dot{K}^\top) / \sqrt{d}. \quad (12)$$

The softmax Jacobian gives, per query row i ,

$$\dot{P}_{ij} = P_{ij} \left(\dot{S}_{ij} - \sum_{j'} P_{ij'} \dot{S}_{ij'} \right), \quad (13)$$

and since $O = PV$,

$$\dot{O} = \dot{P} V + P \dot{V}. \quad (14)$$

Collecting terms per row and defining the four row-wise quantities

$$O_i = \sum_j P_{ij} V_j, \quad A_i = \sum_j P_{ij} \dot{V}_j, \quad (15)$$

$$B_i = \sum_j P_{ij} \dot{S}_{ij} V_j, \quad D_i = \sum_j P_{ij} \dot{S}_{ij}, \quad (16)$$

the output tangent reduces to the compact form

$$\dot{O}_i = A_i + B_i - D_i O_i. \quad (17)$$

Listing 1: Fused forward + JVP attention (Triton, inner loop).

```

1  # one program := (query block pid_m, batch*head
2  )
3  m_i, l_i = -inf, 0.0 # online-softmax max, denom
4  O_i = A_i = B_i = 0.0 # [BLOCK_M, d]
5  primal + tangents
6  D_i = 0.0 # [BLOCK_M]
7  for start_n in range(0, seqlen_k, BLOCK_N):
8  # load q,k,v and tangents tq,tk,tv (fp32)
9  S = dot(q, k) * sm_scale #
10 scores
11 tS = (dot(tq, k) + dot(q, tk)) * sm_scale #
12 score tangent
13 S = mask(S, -inf); tS = mask(tS, 0.0) #
14 padding/causal
15 m_new = maximum(m_i, max(S, axis=1))
16 P = exp(S - m_new) #
17 unnormalized
18 a = exp(m_i - m_new) #
19 rescale old state
20 l_i = a*l_i + sum(P, axis=1)
21 O_i = a*O_i + dot(P, v) #
22 Sum P v
23 A_i = a*A_i + dot(P, tv) #
24 Sum P tv
25 PdS = P * tS
26 B_i = a*B_i + dot(PdS, v) #
27 Sum P tS v
28 D_i = a*D_i + sum(PdS, axis=1) #
29 Sum P tS
30 m_i = m_new
31 0 = O_i / l_i
32 t0 = (A_i + B_i) / l_i - (D_i / l_i) * O # Eq
33 . (jvp-compact)
34 store(0); store(t0)

```

All four quantities are row sums over keys, so they admit the same streaming, $O(N)$ -memory accumulation as the FlashAttention denominator.

A.3 Fused Online-Softmax Algorithm

The kernel keeps FlashAttention’s numerically stable online softmax (running row max m_i and denominator l_i) and accumulates the four unnormalized quantities $(\tilde{O}_i, \tilde{A}_i, \tilde{B}_i, \tilde{D}_i)$ alongside it, rescaling all of them by the same $\exp(m_i^{\text{old}} - m_i^{\text{new}})$ factor whenever the running max updates. After the key loop it normalizes once,

$$O = \tilde{O}/l, \quad \dot{O} = (\tilde{A} + \tilde{B})/l - (\tilde{D}/l) O, \quad (18)$$

which is exactly Equation 17. The $N \times N$ scores and their tangent are never materialized: each key block contributes to the accumulators and is discarded, so peak memory matches standard attention plus the tangent input/output tensors. Listing 1 shows the inner loop; the kernel is autotuned over block sizes keyed on the head dimension, runs on variable-length packed sequences via `cu_seqlens` (matching the sparse latent’s per-sample token counts), loads in bfloat16, and accumulates in fp32.

Listing 2: Autograd bridge and attention patch.

```

1  class FlashAttnVarlenFuncJVP(torch.autograd.
    Function):
2      @staticmethod
3      def forward(ctx, q,k,v, cu_q,cu_k, mq,mk):
4          ctx.save_for_backward(q,k,v); ...
5          return flash_attn_varlen_func(q,k,v, ...)
           # primal
6      @staticmethod
7      def backward(ctx, g):
           # reverse-mode kept
8          with torch.enable_grad():
9              out = flash_attn_varlen_func(q_d,k_d,v_d,
                ...)
10             return torch.autograd.grad(out, [q_d,k_d,
                v_d], g)
11     @staticmethod
12     def jvp(ctx, dq,dk,dv, *a):
           # <- forward-mode
13         out, tout = empty_like(q), empty_like(q)
14         _attn_fwd_jvp_kernel[grid](q,k,v, dq,dk,dv,
            out,tout, ...)
15         return tout
16
17     @contextlib.contextmanager
18     def patch_trellis_attention_for_jvp():
19         orig = flash_attn.flash_attn_varlen_func
20         flash_attn.flash_attn_varlen_func = \
21             lambda q,k,v,*a,**kw:
                FlashAttnVarlenFuncJVP.apply(q,k,v,*a)
22         try: yield
23         finally: flash_attn.flash_attn_varlen_func =
            orig

```

A.4 Integration with Forward-Mode Autodiff

The kernel is exposed to PyTorch through a custom `torch.autograd.Function` whose `forward` calls the original FlashAttention (primal only), whose `backward` keeps standard reverse-mode available (re-running FlashAttention under `enable_grad`), and whose `jvp` static method invokes the fused Triton kernel to return the output tangent. During the tangent pass the backbone’s variable-length attention entry points are monkey-patched to this Function by a context manager and restored on exit, so only the forward-mode pass routes through the kernel and the rest of the pipeline is untouched (Listing 2). At the call site the training step seeds the duals—the latent tangent with $\bar{v}_\phi$ and the scaled-time tangent with the constant 1000—inside a `fwAD.dual_level()` under the patch, then unpacks the dual of the output to obtain g of Equation 11.

A.5 Cost and Numerical Stability

The fused kernel keeps forward-mode differentiation at the memory cost of standard attention: it is $O(N)$ in the token count rather than $O(N^2)$, so it runs at the $N=30,000$ production length where the naive `torch.func.jvp` baseline needs tens of gigabytes and runs out of memory (main paper, JVP-cost table). The overhead is training-time only—inference uses the plain student and never evaluates the tangent. The per-token tangent g is stabilized by the smooth magnitude-preserving cap of the main paper before it enters the self-

consistency target; over the training tokens we profiled we observed no NaN or Inf, and the cap bounds the maximum tangent norm while leaving the median essentially unchanged.

B Additional Qualitative Results

In this section, we present additional qualitative comparisons to further demonstrate the effectiveness of our proposed distillation framework and Channel Match. Figure 3 provides extended visual comparisons on the TRELLIS.2 demonstration assets. We compare our two-step student model against the twelve-step teacher model, as well as the Self-Consistency Match (SCM) and Marginal-Data-Transport (MDT) baselines. Consistent with the main text, our approach recovers geometric details and texture quality that closely approximate the many-step teacher. In contrast, the MDT baseline frequently fractures the underlying geometry, while the SCM baseline preserves geometric completeness but introduces noticeable surface artifacts due to the uncalibrated shape latents.

C Channel Match on TRELLIS.1

To evaluate the generalization of our latent calibration method, we apply Channel Match to the earlier TRELLIS.1 architecture. Unlike TRELLIS.2, TRELLIS.1 operates without a fixed voxel resolution. As illustrated in Figure 4, while the distribution shift induced by few-step sampling in TRELLIS.1 does not manifest as the severe dark or blotchy surface artifacts observed in TRELLIS.2, it nonetheless causes a significant overall degradation in texture quality. Applying our training-free Channel Match accurately restores the per-channel statistics of the few-step shape latents. This correction successfully recovers the texture fidelity produced by the downstream appearance stage, confirming that sampling-induced latent shift is a general phenomenon across different structured-latent pipelines and can be mitigated by our calibration method.

D High-Resolution Generation on TRELLIS.2

We further validate the robustness of Channel Match at a higher generation resolution. Specifically, we evaluate the few-step shape generation on TRELLIS.2 scaled to a 1536^3 grid resolution. As shown in Figure 5, the few-step distribution shift persists at this higher resolution, causing the geometry latent to drift away from the training distribution expected by the frozen PBR texture model and resulting in incorrect material synthesis. The application of Channel Match effectively corrects this shift, eliminating the texture artifacts and restoring the rendering quality without altering the underlying high-resolution geometry. This confirms that our calibration strategy scales reliably to higher-dimensional sparse latents.

References

- Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16251–16261, 2025a.
- Yiwen Chen, Zhihao Li, Yikai Wang, Hu Zhang, Qin Li, Chi Zhang, and Guosheng Lin. Ultra3d: Efficient and high-fidelity 3d generation with part attention. *arXiv preprint arXiv:2507.17745*, 2025b.
- Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26576–26586, 2025c.
- Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 35799–35813, 2023a.
- Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsanit, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b.
- Sebastian Eilermann, René Heesch, and Oliver Niggemann. Conticom-3d: A continuous-time consistency model for 3d point cloud generation. In *International Conference on 3D Vision*, 2026.
- Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14822–14833, 2025.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
- Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE international conference on computer vision (ICCV)*, pages 1501–1510, 2017.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456. pmlr, 2015.
- Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 26565–26577, 2022.
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Fuyun Wang, Huiwen Shi, Xianghui Yang, Qingxiang Lin, Jingwei Huang, Yuhong Liu, et al. Unleashing vecset diffusion model for fast shape generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2523–2533, 2025.
- Yushi Lan, Fangzhou Hong, Shangchen Zhou, Shuai Yang, Xuyi Meng, Yongwei Chen, Zhaoyang Lyu, Bo Dai, Xingang Pan, and Chen Change Loy. Ln3diff++: Scalable latent neural fields diffusion for speedy 3d generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TAPAMI)*, 2025.
- Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TAPAMI)*, 2025.
- Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pages 118582–118600, 2026.
- Shanchuan Lin, Anran Wang, and Xiao Yang. SDXL-Lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2022.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International conference on learning representations (ICLR)*, 2023.
- Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025.

- Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17719–17728, 2025.
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.
- Matteo Pagliardini, Daniele Paliotta, Martin Jaggi, and François Fleuret. Faster causal attention over large sequences through sparse flash attention. *arXiv preprint arXiv:2306.01160*, 2023.
- Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4209–4219, 2024.
- Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.
- Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision (ECCV)*, pages 87–103. Springer, 2024.
- Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, pages 32211–32252, 2023.
- George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Vilecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 3732–3784, 2023.
- Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1798–1808, 2021.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.
- Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision (ECCV)*, pages 1–18. Springer, 2024.
- Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, yajie bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, and Yao Yao. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.
- Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 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 *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 14419–14429, June 2026.
- Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, et al. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27091–27101, 2024.
- Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024a.
- Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 6613–6623, 2024b.
- Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Trans. Graph.*, 43(4), July 2024. ISSN 0730-0301. doi: 10.1145/3658146.
- Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.
- Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

Zanwei Zhou, Taoran Yi, Jiemin Fang, Chen Yang, Lingxi Xie, Xinggang Wang, Wei Shen, and Qi Tian. Few-step flow for 3d generation via marginal-data transport distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence(AAAI)*, volume 40, pages 13853–13861, 2026.



Figure 3: Additional qualitative comparison on TRELLIS.2 demonstration assets. From left to right: input image, the 12-step teacher, SCM @2, MDT @2, and ours @2. Each block displays the PBR render and the untextured mesh. MDT fractures the geometry, SCM leaves surface artifacts, and our method produces results closest to the teacher.

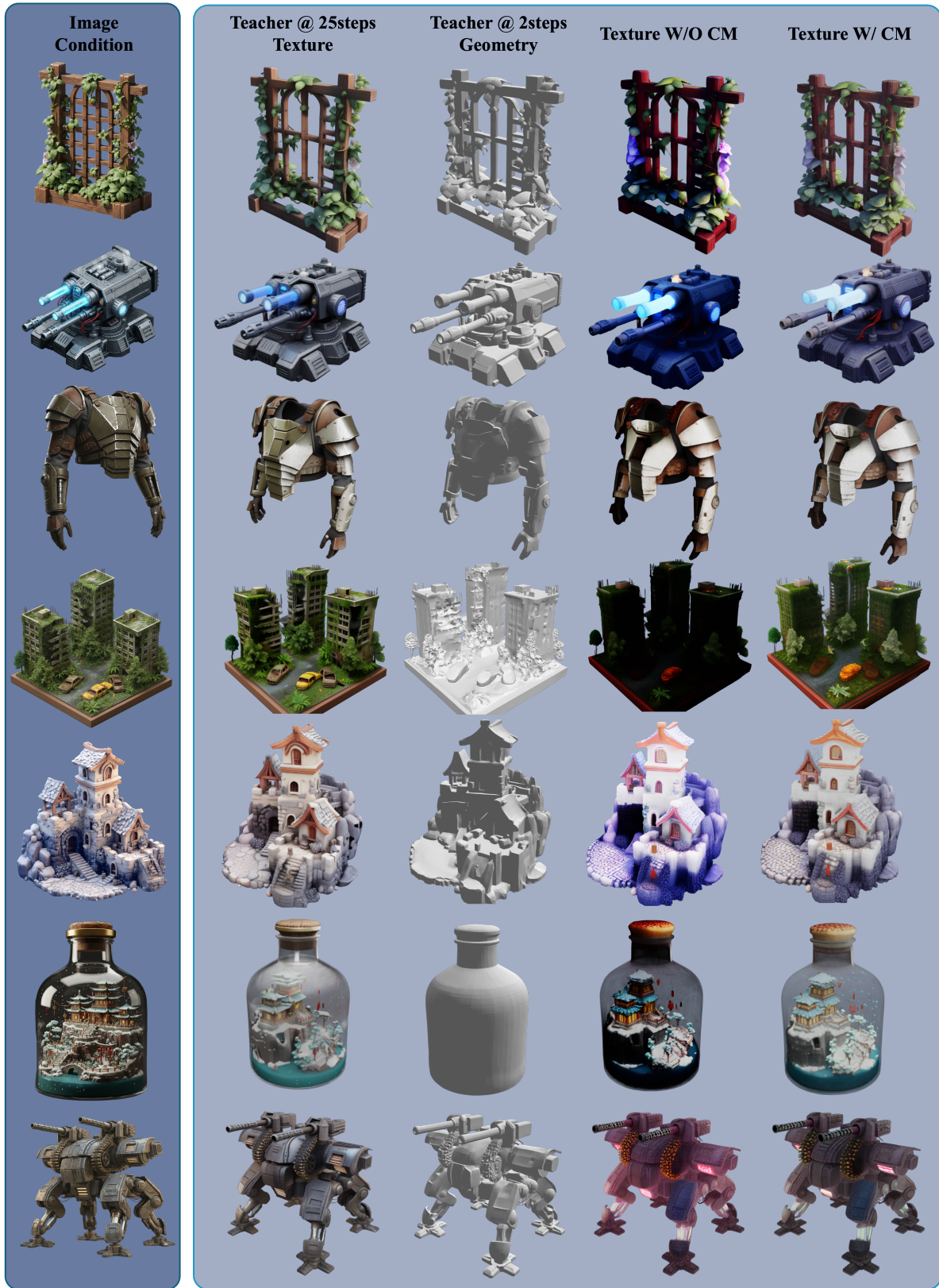


Figure 4: The effect of Channel Match on the TRELLIS.1 architecture (variable voxel resolution). From left to right: input image; the untextured mesh and render of the two-step student without CM; and the mesh and render with CM. Although TRELLIS.1 does not exhibit the severe dark artifacts seen in TRELLIS.2, the few-step sampling still degrades the overall texture quality. Applying CM effectively restores the texture fidelity by calibrating the shifted shape latent.

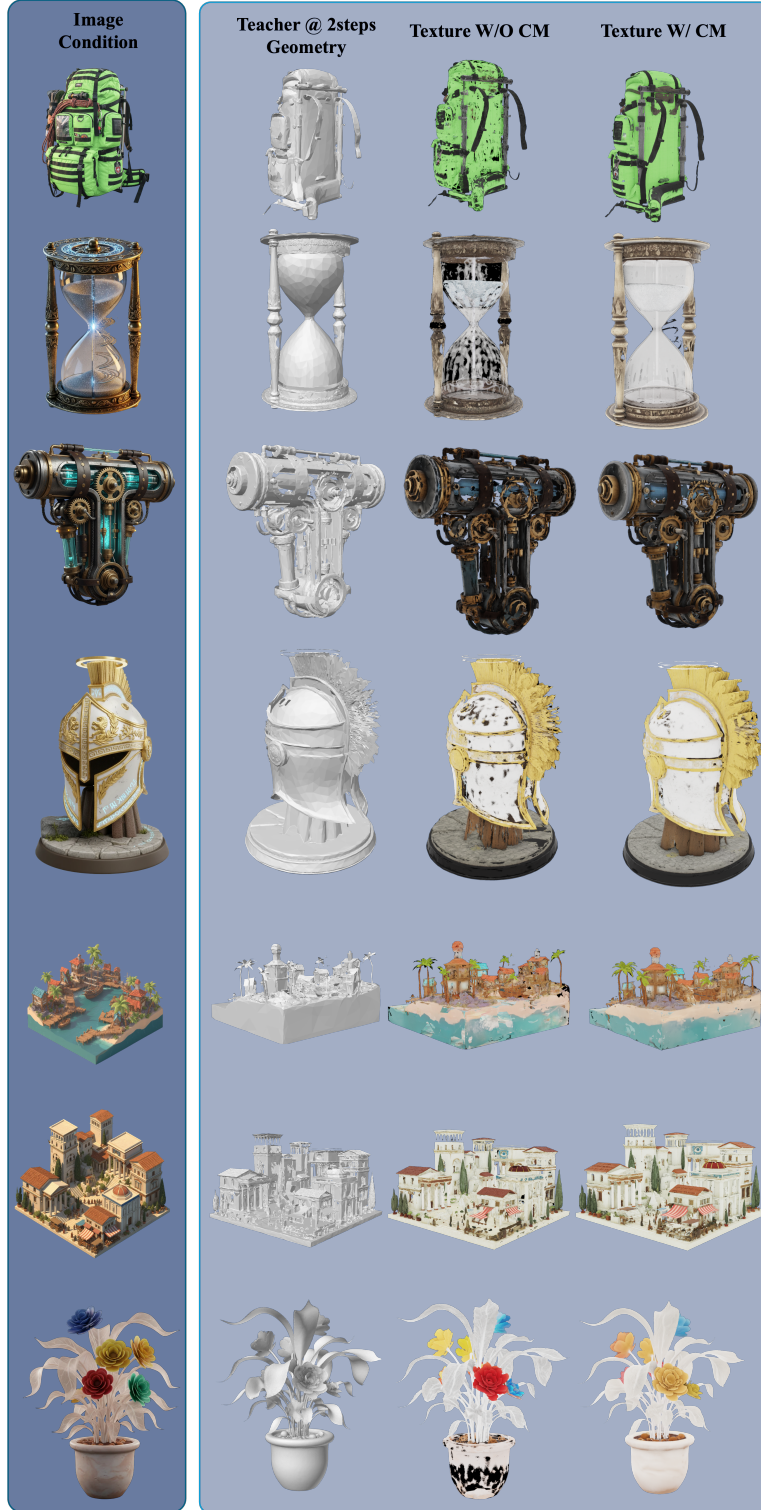


Figure 5: High-resolution (1536^3) shape generation on TRELIS.2. The figure demonstrates the few-step distribution shift and its correction by Channel Match (CM). From left to right: input image; the untextured mesh and PBR render of the two-step student without CM, where the shifted shape latent causes the frozen texture model to produce faulty surface materials (e.g., dark patches); and the corresponding mesh and render with CM applied. Calibrating the latent removes the texture artifacts while maintaining the high-resolution geometry. We consider samples with relatively simple geometry to isolate the effect of latent distribution shift from geometry reconstruction errors, demonstrating that CM effectively corrects the shift at high resolutions.