RETIRE3D — WHERE RESOLUTION LIVES: LOCAL SCALE COMPLETION FOR COARSE-TO-FINE 3D GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Coarse-to-fine 3D generators commonly advance predicted surface regions through a shared finest generation stage. This uniform allocation conflicts with a basic property of 3D geometry: geometric complexity is highly non-uniform in space. Broad, smooth surfaces can often be determined at a coarse scale, whereas thin structures, sharp boundaries, holes, and junctions demand finer precision. The benefit of additional fine-stage computation can therefore vary substantially across regions. We identify this conflict as a mismatch between *uniform computational resolution* and *local geometric demand*, and formulate *Local Scale Completion*: fine-scale computation should follow remaining local geometric demand. We introduce Retire3D, a training-free framework that instantiates this principle through spatially selective evaluation within the high-resolution stage, retaining all latent coordinates. Retire3D uses coarse-stage signals to initialize a fixed evaluation partition, while all fine-scale latent states continue to evolve and provide attention context. History-based propagation supplies unevaluated velocities, and residual checks schedule full reevaluations. The complete spatial representation enters the native downstream pipeline. Experiments on Direct3D-S2 and TRELLIS.2 assess end-to-end efficiency and fidelity to native generation. On Direct3D-S2, Retire3D achieves a native-geometry fidelity score of .927 versus .634 for backbone-adapted Fast3Dcache, a 46.2% relative improvement. With temporal caching, Retire3D achieves a $1.528\times$ end-to-end speedup over native generation on TRELLIS.2. Code is available at `https://anonymous.4open.science/r/Retire3D`.

1 INTRODUCTION

Coarse-to-fine 3D generators commonly advance predicted surface regions toward a shared finest resolution. As spatial representations grow, controlling local computation while preserving interactions across the full representation becomes increasingly important. Modern sparse pipelines make this tractable by building spatial structure and concentrating computation around occupied regions or within spatial and semantic neighborhoods (Ren et al., 2024; Xiang et al., 2025b; Wu et al., 2025; Xiang et al., 2025a). These mechanisms determine where geometry is represented and information exchanged, not where additional resolution remains necessary. Resolution demand is local: a broad wall may already be determined coarsely, while a nearby hole, thin support, sharp edge, or junction may require the finest scale. Once activated, however, both regions generally traverse the same scale transitions. The missing decision is spatial: *where can fine-stage evaluations be reduced while preserving the jointly generated geometry?*

Figure 1 probes this mismatch through a counterfactual replacement experiment. We progressively substitute propagated coarse states for fine regions and measure F-score against the full fine result, using a tolerance of one fine voxel. In the displayed sweep, an oracle ordering by measured refinement debt reaches the 0.98 threshold at approximately 59% substitution, whereas random replacement loses quality much earlier. The spatial example and its insets illustrate how replacement sensitivity varies across a smooth wing and a thin junction. This diagnostic supports a local allocation opportunity: *which* regions receive fine computation matters, not just how many. It is not a timing measurement or an inference-time completion predictor.

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.

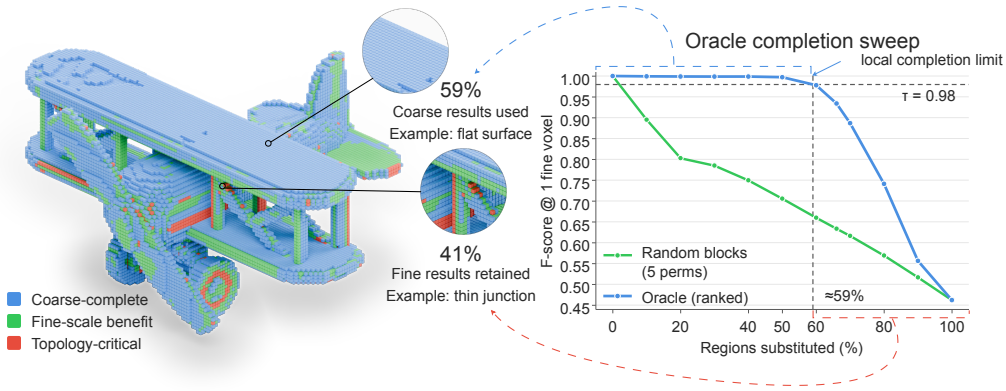

Figure 1: **Fine-scale geometric benefit is spatially non-uniform.** Left: coarse-complete (blue), fine-benefit (green), and topology-sensitive (red) regions. Right: oracle-ordered coarse substitution preserves one-fine-voxel F-score longer than random replacement, reaching $F = 0.98$ at about 59% substitution. This diagnostic is not the deployed score or a measurement of compute savings.

The coarse-to-fine hierarchy supplies an inexpensive prior for allocating this computation: coarse states expose geometric variation, generative change, and structural ambiguity before fine sampling begins. The challenge is to turn these observations into local savings while maintaining coherent joint generation. Regions evaluated less frequently must continue evolving and providing context to the regions still being refined.

We formulate *Local Scale Completion*: fine-scale computation should follow remaining local geometric demand. Retire3D operationalizes this principle through spatially selective evaluation within the high-resolution stage. Selected regions advance through prediction between fresh evaluations, while all high-resolution coordinates and latent states remain in the joint sampling process.

Retire3D is a training-free realization of this principle. Coarse-stage signals initialize a fixed evaluation partition; history-based propagation and residual-triggered reevaluation maintain the complete latent state and its attention context during sampling (Figure 3).

Our contributions are:

- We formulate fine-stage 3D generation as a local computation-allocation problem, motivated by spatially heterogeneous refinement benefit within coarse-to-fine generation.
- We introduce Retire3D, a training-free framework that initializes spatial retirement from coarse-stage evidence, maintains evolving high-resolution states and their contextual participation, and renews evaluations through residual checks.
- We evaluate two coarse-to-fine 3D generators, reporting end-to-end cost, ground-truth reconstruction metrics, and fidelity to native outputs, and examine composition with temporal caching.

2 Related Work

High-resolution 3D generation. Unstructured latent-set representations (Zhang et al., 2023; Zhao et al., 2023) support generators such as CLAY, Hunyuan3D 2.0, and TripoSG (Zhang et al., 2024; Tencent Hunyuan3D Team, 2025; Li et al., 2025). Spatially structured pipelines instead build geometry coarse to fine: XCube uses hierarchical voxel grids; TRELLIS and TRELLIS.2 use structured latents; Direct3D-S2 refines spatial coordinates; and SAM 3D reconstructs geometry, appearance, and layout (Ren et al., 2024; Xiang et al., 2025b;a; Wu et al., 2025; SAM 3D Team et al., 2025). These hierarchies organize geometry while exposing coarse spatial evidence before expensive fine sampling. Direct3D-S2 constructs fine coordinates from a decoded 512^3 surface; TRELLIS.2 obtains fine support through coarse-shape decoder subdivision. In our profiling pilot, high-resolution

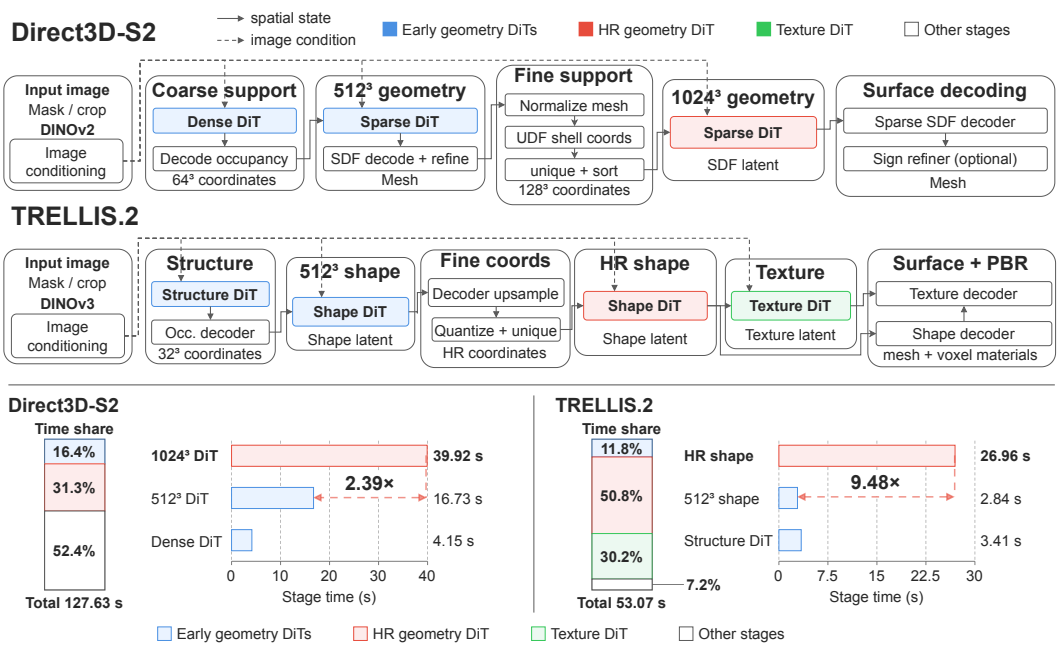

Figure 2: **Native pipelines and stage costs.** Solid arrows pass spatial states; dashed arrows supply image conditioning. Resolutions denote coordinate/output grids. Bottom: mean times and shares over three objects with three runs each. Blue: coarse/low-resolution geometry DiTs; red: high-resolution geometry DiTs; green: texture. High/low ratios are stage-time ratios, not speedups; axes are scaled separately (Appendix B).

geometry DiTs account for 31.3% and 50.8% of their native latency, respectively (Figure 2). Coarse-to-fine generation thus provides both prior information for allocation and a substantial fine-stage cost to target; conditioning, coordinate construction, and decoding remain in the end-to-end measurement.

Efficient 3D generation. Existing work reduces sampling steps, reuses intermediate computation, or accelerates decoding. Distillation enables few-step generation (Salimans & Ho, 2022; Song et al., 2023; Hu et al., 2025); FlashVDM also accelerates volume decoding (Lai et al., 2025). Among training-free methods, Hash3D reuses cross-view/time features (Yang et al., 2025b); Fast3Dcache allocates caching using voxel stability and velocity changes (Yang et al., 2025a); Fast-SAM3D combines modality-aware step caching, spatiotemporal token selection, and decoder aggregation (Feng et al., 2026). The latter methods already exploit heterogeneous computation demands. Our focus is how coarse-stage evidence initializes fine-stage evaluation allocation, and how less-evaluated spatial states continue participating in joint generation.

Temporal reuse and prediction. DDIM and DPM-Solver improve sampling efficiency (Song et al., 2021; Lu et al., 2022), while DeepCache reuses intermediate features (Ma et al., 2024). TeaCache and EasyCache use change estimates to guide reuse (Liu et al., 2025a; Zhou et al., 2025); TaylorSeer forecasts features with finite differences (Liu et al., 2025b), and DiCache uses shallow probes for cache timing and prediction (Bu et al., 2026). These methods exploit redundancy across sampling steps. We consider a complementary dimension: which spatial rows require fresh computation within evaluations that still execute. Spatial selection can therefore be composed with temporal caching, with its benefit depending on the cache schedule and state maintenance.

Spatially adaptive computation. Token merging reduces spatial redundancy (Bolya & Hoffman, 2023); ToCa selectively caches token features (Zou et al., 2025); SADA adapts temporal and spatial sparsity (Jiang et al., 2025); and RALU schedules regional upsampling (Jeong et al., 2026). More closely related, RAS varies regional evaluation frequency, retaining cached K/V and inserting dense steps (Liu et al., 2026). Selective evaluation thus requires not only token selection but also context

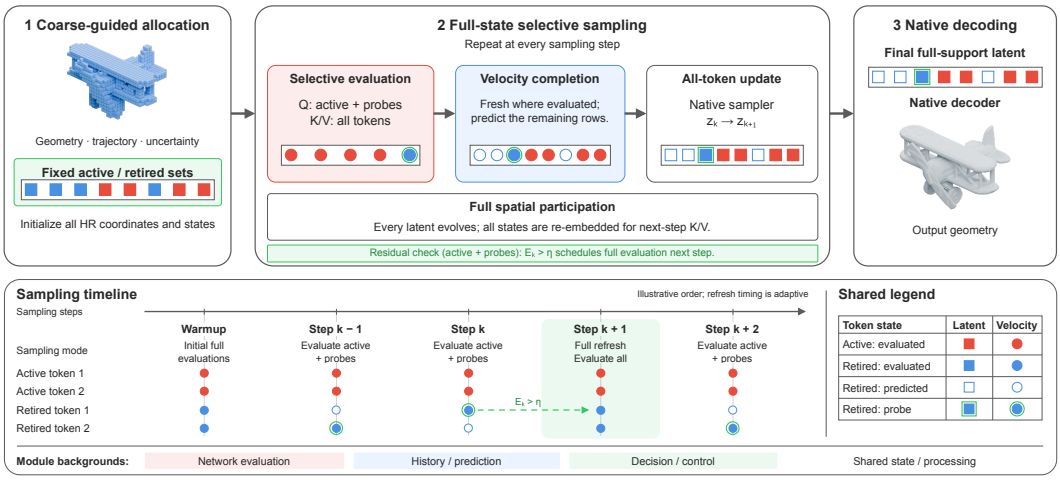

Figure 3: **Retire3D overview.** Coarse evidence initializes a fixed evaluation partition. At each step, fresh and history-predicted velocities jointly advance the complete latent; all regions continue supplying attention context. Residual checks schedule full reevaluations. The final state enters the native decoder. The timeline and airplane renderings illustrate the method.

from unevaluated regions. Retire3D initializes a stage-long partition at the coarse-to-fine boundary and rebuilds embeddings from the evolving full latent to provide approximate context, rather than directly reusing the last-evaluated K/V of retired regions.

From selective computation to local retirement. Prior work shows that regions need not receive equally frequent network evaluations (Zou et al., 2025; Liu et al., 2026). In coarse-to-fine 3D generation, a specific question remains: how can coarse information available before fine sampling guide subsequent computation while maintaining joint shape evolution? Coarse states provide clues about local geometry and generative change, but reducing a region's evaluations does not imply that it stops changing or influencing other regions through attention (Zou et al., 2025). If less-evaluated states stagnate, both their evolution and the context they supply may become inconsistent with the rest of the shape. Local allocation must therefore consider not only where to reduce evaluations, but also how to maintain those regions' states and interactions. We introduce Retire3D as an approach to this local computation-allocation problem in coarse-to-fine 3D generation.

3 Retire3D

Retire3D addresses the local-allocation problem discussed in Section 2 within a native coarse-to-fine sampler. Its central design separates *spatial participation* from *fresh network evaluation*: every high-resolution latent continues to evolve and supply context, while only selected regions receive a fresh network evaluation on ordinary steps. The scale boundary initializes this allocation; subsequent sampling maintains the full state and renews evaluations when the residual monitor triggers. These operations form one sampling loop over the native spatial representation (Figure 3).

3.1 Allocation at the Coarse-to-Fine Boundary

Let Ω contain the N high-resolution token coordinates constructed by the native pipeline. The preceding coarse stage provides an inexpensive starting point for assigning evaluation priority. We map geometric variation G, generative trajectory change C, and surface or subdivision uncertainty U onto these coordinates and combine their normalized values:

$$S_i = w_g \bar{G}_i + w_c \bar{C}_i + w_u \bar{U}_i, \qquad \mathcal{R} = \mathrm{Bottom}_{\rho N}(S), \tag{1}$$

Low-scoring coordinates form the retirement set $\mathcal{R}$; the remainder form the active set $\mathcal{A} = \Omega \setminus \mathcal{R}$. The fraction ρ controls the allocation, and the partition remains fixed within the stage. The score initializes evaluation priority. Active and retired denote evaluation schedules: both sets retain their

fine-scale coordinates, evolve throughout sampling, and participate in attention context. The signals reuse native coarse-stage outputs without an extra network pass; their construction and selection incur a one-time cost. Appendix A gives the mappings and settings.

3.2 Full-State Evolution under Selective Evaluation

Let z_k be the complete latent at sampling time τ_k and $\mathcal{Q}_k$ the coordinates evaluated at that step. An ordinary step uses $\mathcal{Q}_k = \mathcal{A} \cup \mathcal{P}_k$, where $\mathcal{P}_k$ is a small set of retired probes; a full reevaluation uses $\mathcal{Q}_k = \Omega$. The same update rule applies in both cases:

$$\widetilde{v}_{k,i} = \begin{cases} v^{\text{eval}}_{k,i}, & i \in \mathcal{Q}_k, \\ \widehat{v}_{k,i}, & i \notin \mathcal{Q}_k, \end{cases} \qquad z_{k+1} = \text{Step}_{\text{native}}(z_k, \widetilde{v}_k; \tau_k, \tau_{k+1}). \tag{2}$$

Thus retirement changes the source of a token's velocity, while the native sampler advances every token. Implementing this update requires both context for the queried rows and velocities for the unqueried rows.

Context from the evolving state. At each step, we rebuild embeddings from all rows of z_k. Queried rows use native attention connectivity with full spatial K/V membership:

$$O_{\mathcal{Q}_k} = \text{Attn}_{\text{native}}\left(Q_{\mathcal{Q}_k}, K_\Omega, V_\Omega\right). \tag{3}$$

Unqueried rows skip query-side attention and MLP updates. Their input embeddings are held within that forward, and each layer applies its own modulation and K/V projections to those embeddings. After Equation (2), the next step rebuilds them from z_{k+1}. Retired context therefore follows the evolving latent instead of reusing last-evaluated K/V tensors, as in RAS (Liu et al., 2026). These projections preserve contextual participation while approximating the features that full-depth evaluation would produce.

History-based propagation. For an unqueried token i, let a, b index its two latest actual velocity evaluations. Our evaluated configuration uses first-order finite-difference prediction (Liu et al., 2025b; Feng et al., 2026):

$$\widehat{v}_{k,i} = v_{a,i} + \frac{\tau_k - \tau_a}{\tau_a - \tau_b}\left(v_{a,i} - v_{b,i}\right). \tag{4}$$

Only actual evaluations update the history; with fewer than two, the predictor holds the latest evaluated velocity. Zeroth-order holding is also a valid propagation rule and is compared in Section 4.4. In either case, Equation (2) keeps retired latents on an evolving trajectory. This couples the velocities used for their own updates to the state from which their next-step context is constructed.

3.3 Renewing Evaluations during Sampling

The coarse partition sets ordinary-step allocation; fresh evaluations renew the information used to propagate the full state. To determine when to renew it globally, we rotate retired probes across spatial and geometric-risk strata and compare predicted velocities with fresh ones:

$$e_{k,i} = \frac{\|v^{\text{eval}}_{k,i} - v^{\text{pred}}_{k,i}\|_2}{\max(\|v^{\text{eval}}_{k,i}\|_2, \epsilon_k)}, \qquad E_k = \max\{q_{0.9}(e_{k,\mathcal{A}}), q_{0.9}(e_{k,\mathcal{P}_k})\}. \tag{5}$$

Here ϵ_k stabilizes normalization and $q_{0.9}$ is the 90th percentile. For active tokens, a separate monitor predictor uses history renewed only at full refreshes, measuring drift over the refresh interval rather than only since the previous step. Retired probes use their own evaluation histories.

If $E_k > \eta$, the next step evaluates all tokens. A final-step trigger instead causes immediate full reevaluation before committing the update. Warm-up and guidance-mode changes also force full evaluation. Refresh renews histories while preserving the partition. The residual monitors velocity-prediction error; its relationship to the resulting geometry is evaluated empirically in Section 4.4.

Table 1: **Quantitative comparison on 100 Toys4K objects.** We compare reconstruction quality, fidelity to native outputs, and end-to-end efficiency on Direct3D-S2 and TRELLIS.2. Evaluation protocols and sample counts are in Appendices B and B.1.

	Comparability			Fidelity to vanilla			Cost	
Method	CD↓	$F_{.05}$ ↑	vIoU↑	F_{nat} ↑	$C_\times$ ↓	$A_\times \to 1$	Speed↑	Mem (GiB)↓
Direct3D-S2								
vanilla	.0445	72.81	.506	1.000	1.00	1.000	1.000×	20.6
Fast3Dcache	.0452	72.71	.500	.634	67.7	1.241	1.081×	20.7
Fast-SAM3D	.0477	71.03	.485	.242	40.8	1.229	**1.158**×	20.8
RAS	.0454	72.56	.457	.297	235.5	1.406	1.036×	25.6
Retire3D	.0452	72.12	.496	**.927**	**2.26**	**1.023**	1.062×	20.6
TRELLIS.2								
vanilla	.0969	35.29	.169	1.000	1.00	1.000	1.000×	7.7
Fast3Dcache	.0946	37.14	.164	.747	117.6	.688	1.446×	5.8
Fast-SAM3D	.0938	36.44	.149	.082	78.2	.826	**1.767**×	7.1
RAS	.0987	34.73	.161	**.978**	**1.70**	**1.004**	1.125×	12.6
Retire3D	.0970	35.66	.172	.882	23.8	.955	1.225×	7.4

Efficient execution. Query-subset execution, reusable indexing, and reduced data movement realize this loop. Retained context still incurs computation, so fewer queries do not imply proportional runtime savings. The final full-support latent enters the native decoder. End-to-end measurements include selection, propagation, probes, refreshes, and context computation; executed query counts are reported separately.

4 Experiments

We examine local computation allocation at three levels: end-to-end generation, composition with temporal caching, and the state-maintenance and reevaluation choices that govern selective sampling.

4.1 Evaluation Setup

Setting. We accelerate fine geometry on Direct3D-S2 and high-resolution geometry and texture on TRELLIS.2, using their native target settings and a single NVIDIA L40S per job. Baselines are backbone-adapted Fast3Dcache, Fast-SAM3D, and RAS (Yang et al., 2025a; Feng et al., 2026; Liu et al., 2026). All methods share FlashAttention; applicable query-subset methods receive the same kernel implementation. Throughout the main comparison, we use the same Retire3D configuration for all objects: $\rho = .95$, first-order propagation, and residual threshold .30. Settings and adaptation scope appear in Appendix A.

Evaluation. On Toys4K (Stojanov et al., 2021), we select one object from each of 100 categories, one canonical view, and one seed per object. Reconstruction metrics are Chamfer Distance (CD), F-score at 0.05 ($F_{.05}$), and volumetric IoU, following unit-cube normalization and ICP alignment (Xiang et al., 2025b; Wu et al., 2025; Tencent Hunyuan3D Team, 2025). To assess fidelity to the unaccelerated generator, we also compare with the same backbone/seed's native mesh: F_{nat} uses a one-fine-voxel tolerance in fixed coordinates; $C_\times$ and $A_\times$ are component and face-count ratios. Efficiency is end-to-end speedup and peak memory. Timing targets 20 objects stratified by latent size, with interleaved native brackets, medians within objects, and geometric means across objects. Available counts, exclusions, and metric details are in Appendices B and B.1.

4.2 End-to-End Generation

Quantitative results. Ground-truth reconstruction metrics remain close across configurations (Table 1), but fidelity to native generation separates their effects. On Direct3D-S2, Retire3D achieves $F_{\text{nat}} = .927$, versus .634 for backbone-adapted Fast3Dcache, with $C_\times = 2.26$ rather than 40–235. This fidelity-oriented operating point gives 1.062× end-to-end speedup at native memory

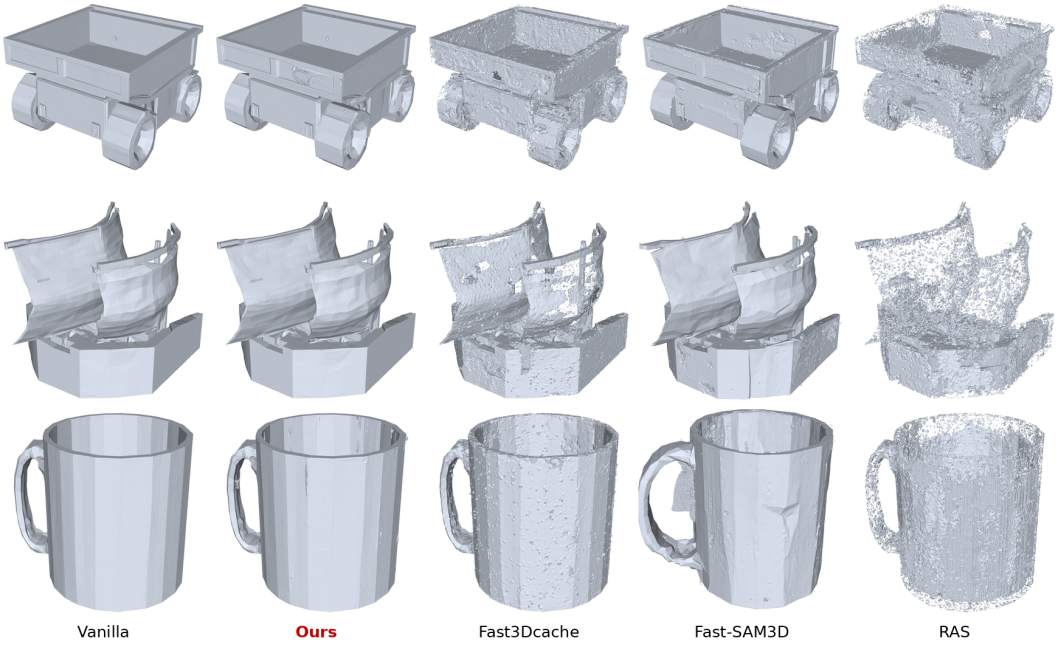

Figure 4: **Direct3D-S2 qualitative comparison.** Three objects from the preselected complexity-ranked set, with identical per-row camera and framing across methods. Full five-object plates for both backbones and the selection rule appear in Appendix E.

Table 2: **Composition with TaylorSeer on TRELLIS.2.** We evaluate quality and efficiency on 50 Toys4K objects. †Speed and memory are reused from Table 1.

	Comparability			Fidelity to vanilla			Cost	
Method	CD↓	$F_{.05}$ ↑	vIoU↑	F_{nat} ↑	$C_{\times}$ ↓	$A_{\times} \to 1$	Speed↑	Mem (GiB)↓
vanilla	.0975	35.29	.181	1.000	1.00	1.000	1.000×	7.4
TaylorSeer (i=2)	.0980	34.85	.176	**.926**	**1.02**	1.022	1.407×	7.5
Retire3D alone†	—	—	—	.887	24.01	.958	1.225×	7.4
TaylorSeer + Retire3D	.1014	33.53	.160	.828	10.26	**1.021**	**1.528**×	**7.4**

usage. On TRELLIS.2, Retire3D reaches 1.225× at 7.4 GiB with $F_{\text{nat}} = .882$. RAS occupies a higher-fidelity, higher-cost point (.978, 1.125×, 12.6 GiB), while Fast3Dcache and Fast-SAM3D offer higher speedups with lower native fidelity. Selective evaluation therefore provides a useful operating point whose balance depends on the backbone's sampling and execution costs.

Qualitative results. Figure 4 shows the corresponding geometry under identical views: Retire3D retains the main shapes and visible fine structures, while the compared outputs exhibit differing amounts of debris and local shape change. Native-output and structural metrics complement these views. Full plates for both backbones, the fixed selection rule, and local failure examples appear in Appendix E.

4.3 Composing with a Step Cache

A temporal cache skips evaluations across steps; Retire3D reduces query rows in evaluations that remain. We combine TaylorSeer (Liu et al., 2025b) at interval 2 with a cache-aligned fixed retirement schedule on TRELLIS.2; this variant replaces residual-triggered scheduling. On the same 50 timed objects, speedup rises from 1.407× to 1.528×, an incremental 8.6% (95% CI [7.2%, 9.9%]), at 7.4 GiB peak memory (Table 2). Executed projection and attention FLOPs also decrease; computation counts and timing checks are in Appendix C.

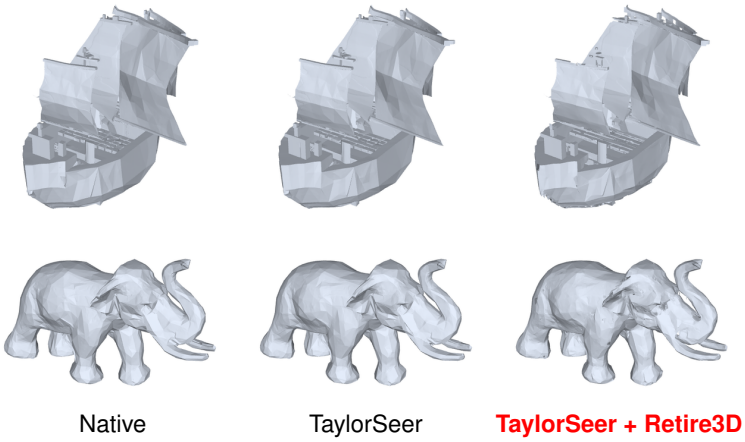

Figure 5: **Visual comparison for the composition in Table 2.** Two selected objects from the 50-object cohort, with identical views and framing across methods. Columns: native, TaylorSeer, and TaylorSeer plus Retire3D. Additional examples and per-object metrics appear in Appendix E.

Table 3: **Refresh ablation.** Fifty held-out Direct3D-S2 objects; speedup S uses four of them. $D_A = |A_\times - 1|$; Q: executed query fraction. Full results: Appendix D.

Variant	$F_{\text{nat}} \uparrow$	$C_\times \downarrow$	$D_A \downarrow$	Q	$S \uparrow$
No extra refresh	.086	250.00	3.395	.540	1.30
Matched count, late refreshes	.609	17.03	.333	.788	1.12
Matched count, uniform	.881	5.22	.032	.788	1.14
Residual-triggered (reference)	.929	2.01	.014	.790	1.14

Table 4: **Retired-state ablation.** Fifty held-out Direct3D-S2 objects in a separate, untimed sampler. $D_A = |A_\times - 1|$; Q: executed query fraction. Compare scores within this table. Full results: Appendix D.

Variant	$F_{\text{nat}} \uparrow$	$C_\times \downarrow$	$D_A \downarrow$	Q
Update only at refresh	.522	22.21	.204	.642
Frozen retired states	.446	6.36	.063	.642
Full-depth K/V, frozen states	.458	3.15	.047	1.000
Embedding K/V + velocity hold	.869	4.94	.033	.642

The extra acceleration trades fine fidelity for lower cost: F_{nat} changes from .926 to .828 and $C_\times$ from 1.02 to 10.26, while the face ratio remains similar (1.022 vs. 1.021). Together, the paired timings and executed-work counts show that spatial evaluation reduction can operate alongside temporal caching. Figure 5 illustrates the retained overall shapes and local geometric changes in two selected examples. Schedule sensitivity and local structural changes are documented in Appendices C and E.

4.4 What Governs Selective Evaluation?

We separate the requirements of the sampling process from choices within its implementation. The main ablation study uses fifty held-out objects per backbone, with timing on four large objects. Tables 3 and 4 report the Direct3D-S2 ablations; complete results, sample counts, and sampler differences appear in Appendix D.

Refresh density and placement. Our refresh rule renews full-token evaluations when the monitored prediction residual exceeds the threshold, alongside mandatory warm-up and guidance-change evaluations (Section 3.3). On Direct3D-S2 (Table 3), omitting discretionary refresh gives $F_{\text{nat}} = .086$; distributing the matched number of refreshes throughout sampling gives .881, compared with .609 when concentrating them late. Residual-triggered placement raises fidelity to .929 and reduces $C_\times$ from 5.22 to 2.01, at nearly equal query cost and the same rounded speedup ($1.14\times$). On TRELLIS.2, residual and matched uniform schedules trade fidelity (.869 vs. .893) against component ratio (25.62 vs. 32.82); every-two-step refresh reaches .941 with about 6% more queries

and lower speedup ($1.28\times$ vs. $1.35\times$). These comparisons assess the quality–cost effects of refresh frequency and placement under the evaluated settings.

Maintaining the full state. In the separate Direct3D-S2 sampler of Table 4, embedding-based K/V with per-step velocity holding reaches $F_{\text{nat}} = .869$, versus .446–.522 at the same query fraction when retired context is supplied only at refresh steps. Full-depth K/V with frozen retired latents reaches .458 at $Q = 1$. These comparisons support the joint role of state propagation and continued contextual participation, rather than isolating a K/V choice in the main sampler. The corresponding alternatives are unavailable on TRELLIS.2.

Allocation and propagation policies. The main comparison uses a fixed reference configuration. Controlled ablations show that velocity holding improves native fidelity over first-order propagation, while the effect of coarse-priority allocation relative to random partitions depends on the backbone. Appendix D reports these controls and the retirement-rate sweep, distinguishing the performance of the complete method from the merits of individual policy choices.

5 Conclusion

We study local computation allocation within coarse-to-fine 3D generation. Retire3D uses coarse-stage evidence to initialize a fixed allocation of high-resolution evaluations. On ordinary sampling steps, active tokens and rotating probes receive fresh evaluations, while retired tokens advance through history-based propagation and are re-embedded to supply attention context. Residual checks schedule full reevaluations, and the complete latent representation is retained for native decoding. On Direct3D-S2, Retire3D achieves a native-geometry fidelity score of .927 versus .634 for backbone-adapted Fast3Dcache, a 46.2% relative improvement. Combined with TaylorSeer on TRELLIS.2, it achieves a $1.528\times$ end-to-end speedup over native generation, with a trade-off in fine geometric fidelity. The diagnostics highlight state maintenance and refresh placement as central considerations, with allocation and propagation policies controlling the resulting quality–cost balance.

AI use statement

Large language models assisted with literature discovery, drafting, translation, language editing, code inspection, and analysis. The authors reviewed the AI-assisted content and take full responsibility for the final work.

References

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4599–4603, 2023.

Jiazi Bu, Pengyang Ling, Yujie Zhou, Yibin Wang, Yuhang Zang, Dahua Lin, and Jiaqi Wang. DiCache: Let diffusion model determine its own cache. In *International Conference on Learning Representations (ICLR)*, 2026.

Weilun Feng, Mingqiang Wu, Zhiliang Chen, Chuanguang Yang, Haotong Qin, Yuqi Li, Xiaokun Liu, Guoxin Fan, Zhulin An, Libo Huang, Yulun Zhang, Michele Magno, and Yongjun Xu. Fast-SAM3D: 3dfy anything in images but faster. In *International Conference on Machine Learning (ICML)*, 2026.

Hanzhe Hu, Tianwei Yin, Fujun Luan, Yiwei Hu, Hao Tan, Zexiang Xu, Sai Bi, Shubham Tulsiani, and Kai Zhang. Turbo3D: Ultra-fast text-to-3D generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23668–23678, 2025.

Wongi Jeong, Kyungryeol Lee, Hoigi Seo, and Se Young Chun. Training-free mixed-resolution latent upsampling for spatially accelerated diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18673–18682, 2026.

Ting Jiang, Yixiao Wang, Hancheng Ye, Zishan Shao, Jingwei Sun, Jingyang Zhang, Zekai Chen, Jianyi Zhang, Yiran Chen, and Hai Li. SADA: Stability-guided adaptive diffusion acceleration. In *International Conference on Machine Learning (ICML)*, 2025.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Fuyun Wang, Huiwen Shi, Xianghui Yang, Qingxiang Lin, Jingwei Huang, Yuhong Liu, Jie Jiang, Chunchao Guo, and Xiangyu Yue. Unleashing vecset diffusion model for fast shape generation. *arXiv preprint arXiv:2503.16302*, 2025.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao. TripoSG: High-fidelity 3D shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7353–7363, 2025a.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with TaylorSeers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Ziming Liu, Yifan Yang, Chengruidong Zhang, Yiqi Zhang, Lili Qiu, Yang You, and Yuqing Yang. Region-adaptive sampling for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. DeepCache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15762–15772, 2024.

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 (CVPR)*, 2024.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.

SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J. Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. SAM 3D: 3Dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023.

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 (CVPR)*, 2021.

Tencent Hunyuan3D Team. Hunyuan3D 2.0: Scaling diffusion models for high resolution textured 3D assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Philip Torr, Xun Cao, and Yao Yao. Direct3D-S2: Gigascale 3D generation made easy with spatial sparse attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 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. *arXiv preprint arXiv:2512.14692*, 2025a.

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

Mengyu Yang, Yanming Yang, Chenyi Xu, Chenxi Song, Yufan Zuo, Tong Zhao, Ruibo Li, and Chi Zhang. Fast3Dcache: Training-free 3D geometry synthesis acceleration. *arXiv preprint arXiv:2511.22533*, 2025a.

Xingyi Yang, Songhua Liu, and Xinchao Wang. Hash3D: Training-free acceleration for 3D generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21481–21491, 2025b.

Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3DShape2VecSet: A 3D shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics*, 42(4), 2023. doi: 10.1145/3592442.

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. *arXiv preprint arXiv:2406.13897*, 2024.

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3D shape generation based on shape-image-text aligned latent representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Xin Zhou, Dingkang Liang, Kaijin Chen, Tianrui Feng, Xiwu Chen, Hongkai Lin, Yikang Ding, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching. *arXiv preprint arXiv:2507.02860*, 2025.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In *International Conference on Learning Representations (ICLR)*, 2025.