MANY STEPS, ONE PASS: LATENT TRAJECTORY SIMULATION FOR EFFICIENT LIDAR SCENE COMPLETION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion-based LiDAR scene completion unlocks the potential to transform sparse scans into rich, detailed 3D geometry. However, traditional iterative denoising requires invoking the full denoising backbone at every step, burdening inference with substantial computational cost. While some distillation methods aims to streamline this process, most of which reduces the nuanced evolution of reconstruction to a single endpoint, discarding the valuable intermediate states. To achieve a better trade-off between accuracy and efficiency, we introduce *LiTE* (**Li**DAR **T**rajectory **E**volution), which performs multiple teacher-aligned latent updates at a deep feature interface within each student transition, decoupling internal geometric evolution from repeated full-backbone evaluation. Within this framework, we propose Visibility-Conditioned Latent Transport (VCLT) to execute these latent updates while maintaining robust local scan conditioning as the trajectory unfolds; it dynamically updates scan-to-point correspondences at the feature level and leverages sensor-ray visibility to precisely guide geometric refinement. Comprehensive experiments on SemanticKITTI and KITTI-360 shows that LiTE achieves state-of-the-art Chamfer distance and JSD, while delivers an $18.4\times$ inference speedup over the 50-step LiDiff teacher and $5.9\times$ improvement over 8-step ScoreLiDAR.

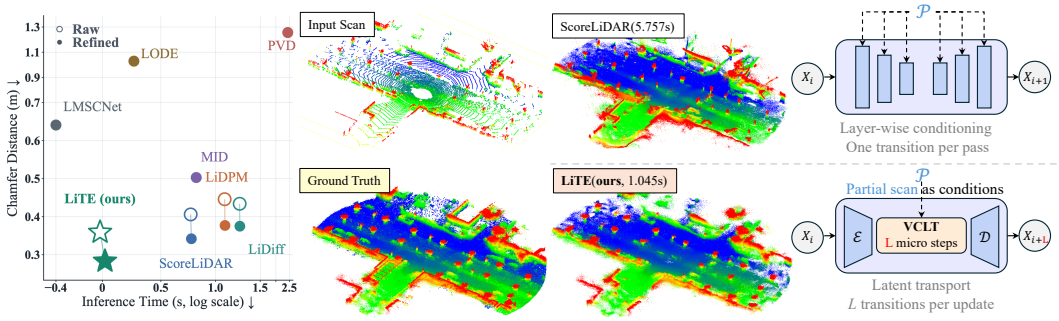


Figure 1: Overview of LiTE. The left panel compares Chamfer distance against inference time, distinguishing raw outputs from refined ones; points closer to the **lower-left corner** indicate better quality and faster inference. The middle panel shows qualitative completions for the input scan, ground truth, ScoreLiDAR, and LiTE. The right panel contrasts layer-wise conditioning, which performs one transition per backbone pass, with LiTE, which inserts VCLT between the encoder and decoder to execute L teacher-aligned micro steps in one pass.

1 INTRODUCTION

LiDAR scene completion aims to reconstruct missing scene geometry from sparse observations. By recovering unobserved regions, it provides a more complete representation for downstream perception and planning (Song et al., 2017; Roldao et al., 2020; Vizzo et al., 2022). Recent diffusion-based approaches have demonstrated promising performance by progressively denoising point sets conditioned on the observed scan, yielding detailed outdoor scene geometry (Nunes et al., 2024; Martyniuk et al., 2025). In particular, LiDiff (Nunes et al., 2024) incorporates local scan features by associating generated points with nearby observations during denoising. However, its iterative

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.

denoising process incurs substantial inference latency, limiting the practical efficiency of diffusion-based LiDAR completion. This has sparked growing interest in accelerating diffusion-based LiDAR completion while retaining geometric fidelity.

Among existing approaches to diffusion acceleration, model distillation has proven effective in reducing iterative generation to only a few denoising steps (Salimans & Ho, 2022; Song et al., 2023; Kim et al., 2024). CTM (Kim et al., 2024) already provides a parameterization for transitions over arbitrary time intervals. ScoreLiDAR (Zhang et al., 2025) brings this idea to LiDAR scene completion by distilling a pretrained LiDiff model (Nunes et al., 2024) into a few-step student. However, its distillation objective primarily transfers teacher knowledge through the completed output, without explicitly modeling the intermediate geometric states along LiDiff's denoising process. For LiDAR completion, these intermediate states are particularly important because point motion continuously changes the spatial relationship between generated geometry and the observed scan. Building on arbitrary-time transition parameterization, LiTE focuses on how a long transition is executed internally: instead of representing it only through its output, it evolves geometry-carrying latent query states, supervises them with intermediate teacher states, and updates scan conditioning as the queries move. The key distinction is therefore the internal realization of a long transition, rather than long-transition prediction alone. We therefore ask the following question:

> Can a long student transition preserve and execute multi-step trajectory evolution within one backbone evaluation, while maintaining geometric fidelity?

In search of an answer, we propose *LiTE* (**Li**DAR **T**rajectory **E**volution), which executes each long student transition as latent trajectory simulation. Rather than predicting only its endpoint, LiTE evolves teacher-aligned latent query states at a deep U-Net interface. Transition consistency and teacher-state supervision train this internal trajectory, while ground-truth alignment anchors the terminal prediction. To execute these updates without repeated backbone evaluation, Visibility-Conditioned Latent Transport (VCLT) recurrently transports the latent state through multiple teacher-aligned micro steps while the surrounding encoder and decoder are evaluated once. At each micro step, VCLT refreshes scan-to-point correspondences and uses sensor-ray visibility to route local versus global motion, suppressing misleading guidance from occluded returns. Terminal structural supervision acts as an auxiliary constraint on the final spatial distribution and local geometry.

We conduct comparative experiments on SemanticKITTI and KITTI-360 (Behley et al., 2019; Liao et al., 2023), together with ablation studies of our key components. As shown in Figure 1, LiTE improves completion quality over representative baselines while using fewer full backbone evaluations, and achieves, on SemanticKITTI, an $18.4\times$ sampling speedup over the 50-step LiDiff teacher and a $5.9\times$ speedup over 8-step ScoreLiDAR. Together, these results show that LiTE decouples multi-step trajectory evolution from repeated backbone evaluation while preserving completion quality and improving it over representative baselines.

Our main contributions are summarized as follows:

- **Teacher-aligned latent execution in one forward pass.** We formulate efficient LiDAR completion as latent trajectory simulation, executing multiple teacher-aligned latent updates within a single student forward pass and decoupling internal geometric evolution from repeated full-backbone evaluation.
- **Visibility-conditioned latent transport.** We develop VCLT to maintain geometric conditioning along the simulated latent trajectory by updating scan-to-point associations as the latent state evolves and regulating local observation-driven motion through sensor-ray visibility. Terminal structural supervision provides an auxiliary constraint on the spatial distribution and local geometry of the final completion.
- **Strong quality–efficiency trade-off.** Trained on only 5.2% of LiDiff's training set for 20 epochs, one checkpoint trained on SemanticKITTI is evaluated directly on both SemanticKITTI and KITTI-360, without dataset-specific retraining or checkpoint selection. It supports multiple inference depths without retraining and runs $18.4\times$ and $5.9\times$ faster than the 50-step LiDiff teacher and 8-step ScoreLiDAR on SemanticKITTI.

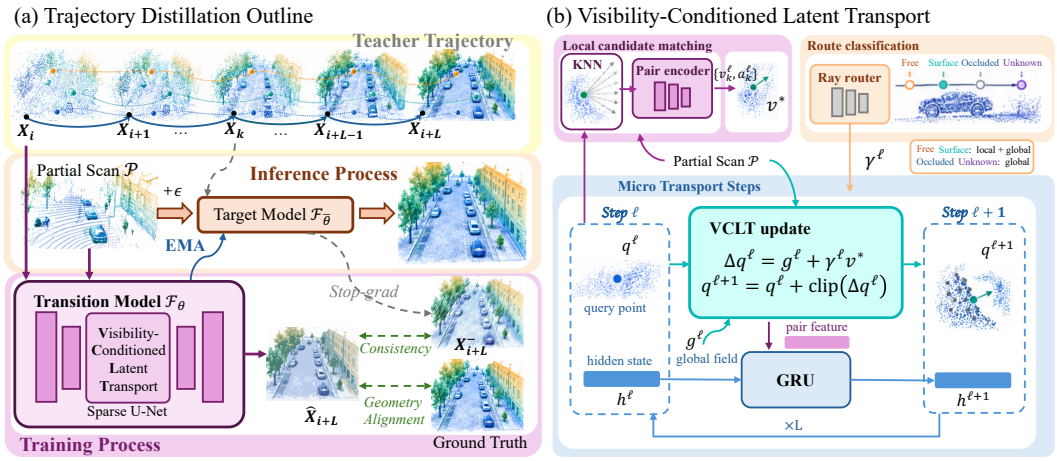


Figure 2: LiTE. (a) A sparse U-Net trains long transitions with teacher trajectory supervision and terminal geometry constraints. (b) VCLT executes the internal micro steps, refreshing candidate matches and ray routes at each step, combines gated local motion with a global field, and propagates a GRU hidden state. The EMA model is used at inference.

2 The LiTE Framework

2.1 Overview and Formulation

Let $P \in \mathbb{R}^{N_p \times 3}$ be a partial scan and $Y \in \mathbb{R}^{N \times 3}$ its complete scene. A conditional diffusion teacher recovers geometry through repeated denoising updates,

$$X_{i+1} = \mathcal{A}_i(X_i, D_\phi(X_i, \tau_i; P)), \qquad i = 0, \dots, M-1, \tag{1}$$

where D_ϕ is the denoiser, $\mathcal{A}_i$ the sampling update, and $\tau_0 > \cdots > \tau_M = 0$.

Given a pretrained LiDiff teacher (Nunes et al., 2024), our goal is to convert it into an efficient student, rather than training a new model from scratch. Following CTM (Kim et al., 2024), we parameterize student transitions $F_\theta(X, t, s; P)$, $s \le t$, over arbitrary teacher-time intervals. An endpoint-only realization of these intervals, however, does not explicitly preserve the intermediate geometric evolution along the teacher trajectory. LiTE therefore focuses on the internal execution of each transition, evolving its latent query state through teacher-aligned micro-steps while the surrounding backbone is evaluated once.

Figure 2 summarizes our solution. Figure 2(a) shows how LiTE trains long student transitions along the teacher trajectory. To execute the skipped evolution, LiTE performs teacher-aligned latent micro-steps at a deep U-Net interface, as illustrated in Figure 2(b). The encoder and decoder are evaluated once per student transition; the intermediate trajectory is instead simulated through lightweight latent updates at the shared deep interface. We refer to this design as latent trajectory simulation. As the latent point state evolves, its relationship with the observed scan changes, so previously established scan-to-point associations can become unreliable. Recomputing correspondences by proximity alone does not fully resolve this issue because an occluded point may be closest to a foreground return. We therefore introduce Visibility-Conditioned Latent Transport (VCLT) to update local correspondences along the latent trajectory and regulate their geometric guidance using sensor-ray visibility. The following subsections detail each component of LiTE.

2.2 Long-Transition Distillation

We first define the endpoint and consistency supervision that transfers teacher trajectory information to each long student transition. Figure 2(a) summarizes this teacher-trajectory supervision and terminal alignment. Given states sampled from the same recorded teacher trajectory, the student is trained to reach a consistent destination from different starting states while remaining anchored to the corresponding LiDiff endpoint. The internal micro-step execution and its intermediate supervision are introduced in Section 2.3.

Transition parameterization. A sparse U-Net predicts a transition displacement $\Delta_\theta(X, t, s; P)$, conditioned on both the starting time t and destination time s. Following CTM (Kim et al., 2024), for $s < t$, we parameterize

$$F_\theta(X, t, s; P) = X + (1 - \rho_{t,s})\Delta_\theta(X, t, s; P), \qquad \rho_{t,s} = \frac{\sigma_s}{\sigma_t}, \tag{2}$$

with $F_\theta(X, t, t; P) = X$, where σ_t follows the teacher noise schedule. This parameterization allows a single student model to represent transitions of different temporal spans.

EMA endpoint consistency. To make long transitions consistent across different starting states, we sample $i < k \le j$ from the same teacher trajectory, with $\tau_i > \tau_k \ge \tau_j$. The online model maps the earlier state X_i to the destination time τ_j, while an EMA copy starts from X_k and provides a stop-gradient target at the same destination:

$$\widehat{X}_j = F_\theta(X_i, \tau_i, \tau_j; P), \qquad X_j^- = \mathrm{sg}[F_{\bar{\theta}}(X_k, \tau_k, \tau_j; P)]\,, \tag{3}$$

When $k = j$ the identity boundary reduces X_j^- to the teacher state X_j.

The LiDiff trajectory preserves particle identities, allowing the predicted endpoint to be compared with the destination of each source particle. Particle-wise alignment alone, however, does not distinguish unsupported predictions from uncovered target regions. We therefore combine three complementary terms for a prediction A and destination B:

$$\mathcal{S}_{\boldsymbol{\alpha}}(A, B) = \alpha_{\mathrm{a}} \underbrace{\ell_{\mathrm{H}}(A, B)}_{\substack{\text{particle-aligned}\\ \text{endpoint loss}}} + \alpha_{\mathrm{p}} \underbrace{\ell_{\mathrm{H}}(A, \mathrm{NN}(A, B))}_{\substack{\text{prediction-to-target}\\ \text{proximity}}} + \alpha_{\mathrm{c}} \underbrace{\ell_{\mathrm{H}}(\mathrm{NN}(B, A), B)}_{\substack{\text{target-to-prediction}\\ \text{coverage}}}, \tag{4}$$

where ℓ_{H} is the smooth L_1 loss and $\mathrm{NN}(A, B)$ assigns each point in A its nearest neighbor in B. Nearest-neighbor assignments retain gradients for the selected prediction coordinates. With component weights $\boldsymbol{\lambda}^{\text{agreement}}$, the EMA agreement loss is $\mathcal{L}_{\text{agreement}} = \mathcal{S}_{\boldsymbol{\lambda}^{\text{agreement}}}(\widehat{X}_j, X_j^-)$.

Direct teacher supervision. The EMA target changes throughout training and may deviate from the recorded teacher trajectory. We therefore additionally supervise the prediction against the fixed teacher state X_j using the same point-set objective, augmented by agreement with teacher displacement directions and magnitudes. This defines $\mathcal{L}_{\text{teacher}}$ and anchors the learned transition to the recorded trajectory. Together, EMA agreement and direct teacher supervision define the trajectory consistency objective $\mathcal{L}_{\text{consistency}} = \lambda_{\text{agreement}}\mathcal{L}_{\text{agreement}} + \lambda_{\text{teacher}}\mathcal{L}_{\text{teacher}} + \lambda_{\text{endpoint}}\mathcal{L}_{\text{endpoint}}$.

For terminal transitions, the same hybrid-cloud objective against complete scene Y is included as $\mathcal{L}_{\text{endpoint}}$ in $\mathcal{L}_{\text{consistency}}$, constraining proximity and coverage; structural losses handle higher-order spatial distribution and local geometry.

2.3 Visibility-Conditioned Latent Transport

VCLT is LiTE's mechanism for executing the internal latent trajectory introduced in Section 2.1. Figure 2(b) illustrates how the latent query state is recurrently transported while candidate matches and ray routes are refreshed. For a long transition from X_i to X_j, VCLT performs $L = j - i$ teacher-aligned micro-steps at a deep U-Net interface before the bottleneck; we refer to L as the transition's interval span. Across different numbers of outer transitions, this teacher-aligned micro-step count remains fixed; the transitions only repartition the same total workload. The encoder and decoder are evaluated once for the entire student transition, while VCLT evolves the internal query state across the L teacher intervals through lightweight recurrent transport. At each micro-step ℓ, $q^{(\ell)}$ denotes the current query location and $h^{(\ell)}$ the hidden state that carries trajectory history.

At the deep interface, each query $q_m^{(0)}$ is a sparse feature-token location whose feature initializes $h_m^{(0)}$. VCLT moves these matching locations and recurrent states while the sparse tensor lattice remains fixed. For source particle $x_{i,n}$, let $\pi(n) = \arg\min_m \|x_{i,n} - q_m^{(0)}\|_2$; its lifted position is

$$\widehat{x}_n^{(\ell)} = x_{i,n} + q_{\pi(n)}^{(\ell)} - q_{\pi(n)}^{(0)}. \tag{5}$$

The displacement supplies particle-level supervision, while $h^{(L)}$ returns to the fixed bottleneck feature stream for decoding.

Visibility-conditioned ray routing. At each micro-step, VCLT first evaluates whether local scan evidence is geometrically valid for the current query state. We project P to a spherical range image and keep the nearest observed range r^* on each valid ray. For a query $q^{(\ell)}$ with range $r = \|q^{(\ell)}\|_2$, let $d = r - r^*$. A margin m and temperature η produce soft free and occluded weights

$$w_{\text{free}} = \sigma\left(\frac{-d-m}{\eta}\right), \quad w_{\text{occ}} = \sigma\left(\frac{d-m}{\eta}\right), \quad w_{\text{surf}} = (1-w_{\text{free}})(1-w_{\text{occ}}), \tag{6}$$

The weights are normalized after masking invalid rays, while queries without valid rays are assigned $w_{\text{unk}} = 1$. Free space queries also obtain an outward cue $u = (r^* - r)_+ q^{(\ell)}/r$, which is clipped to a bounded range.

These visibility states determine how local observations contribute to point motion. Surface and free-space queries can exploit local scan evidence together with the global motion field, whereas occluded and unknown queries rely primarily on the global field to avoid attraction toward foreground surfaces.

Visibility-conditioned latent update. Given the current query $q^{(\ell)}$, VCLT retrieves its K nearest encoded scan candidates. A pair encoder scores these candidates and predicts local displacements $v_k^{(\ell)}$ using scan features, relative geometry, hidden state, visibility information, and the current teacher interval. Let k_ℓ^* denote the selected candidate. In parallel, a candidate-independent global motion $g^{(\ell)}$ is predicted from query history and pooled scan context. We define the local guidance weight as $\gamma^{(\ell)} = w_{\text{surf}}^{(\ell)} + w_{\text{free}}^{(\ell)}$. The query state is updated by

$$q^{(\ell+1)} = q^{(\ell)} + \text{clip}\left(g^{(\ell)} + \gamma^{(\ell)} v_{k_\ell^*}^{(\ell)}\right). \tag{7}$$

This update combines a global trajectory prior with visibility-aware local geometric guidance. Candidate selection is discrete in the forward pass, while softmax-based gradients are used during training. After each update, a GRU propagates the hidden state $h^{(\ell)}$ using the selected pair features and predicted displacement. The updated query location is then used to recompute scan candidates and visibility states for the next micro-step. After the final micro-step, the resulting latent state is injected back into the U-Net and propagated through the remaining layers to produce the final transition output.

Latent-trajectory supervision. The latent micro-steps are supervised by the corresponding intermediate states of the LiDiff teacher trajectory. Since VCLT operates on sparse query locations rather than the full particle set, we transfer the query displacement back to the source particles to obtain an intermediate predicted cloud $\hat{\boldsymbol{X}}^{\boldsymbol{\ell}}$. This cloud is compared with the corresponding teacher state $X_{i+\ell}$. We supervise three aspects of the latent evolution: intermediate position agreement, consistency with teacher motion direction, and free-space motion constraints: $\mathcal{L}_{\text{VCLT}} = \lambda_{\text{position}}\mathcal{L}_{\text{position}} + \lambda_{\text{direction}}\mathcal{L}_{\text{direction}} + \lambda_{\text{freespace}}\mathcal{L}_{\text{freespace}}$. The VCLT target is teacher-aligned to the corresponding teacher states, rather than an exact reproduction of the teacher trajectory. The detailed particle assignment and component definitions are provided in Appendix D.

2.4 Structural Supervision and Few-Step Inference

Long-transition distillation and latent-trajectory supervision constrain the teacher-aligned trajectory and its endpoint. Here Z denotes the raw terminal student output before optional refinement. We use terminal structural supervision as an auxiliary regularizer for aspects not fully specified by point-level alignment, namely the scene-level spatial distribution and local geometric organization of the final completion.

As illustrated in Figure 3, this auxiliary objective consists of two complementary groups. The spatial distribution constraints compare the terminal prediction Z with the complete ground truth Y through multiscale occupancy and density. Occupancy alignment encourages accurate scene support, while density consistency reduces excessive point concentration in occupied regions. The

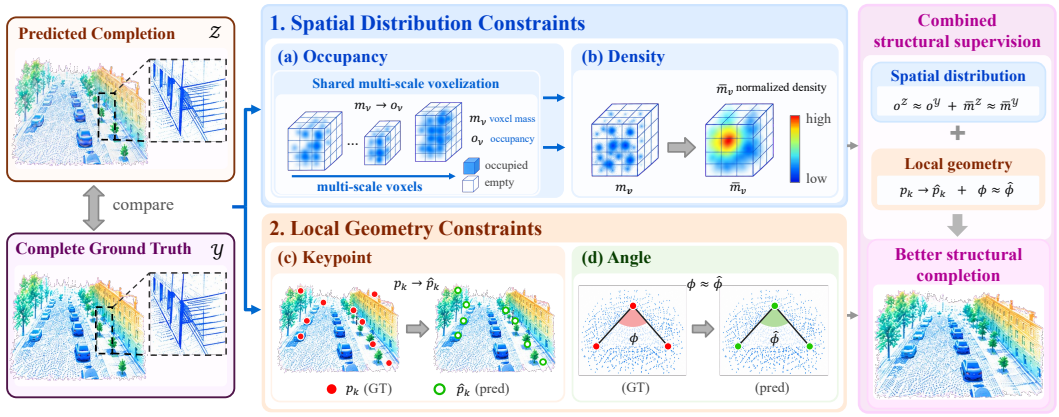


Figure 3: Structural supervision compares the terminal completion Z with complete ground truth Y. (a,b) Multiscale occupancy and density constraints align occupied support and penalize point concentration. (c,d) Saliency-weighted keypoint coverage and triplet-angle matching preserve salient locations and local corner geometry.

local geometry constraints preserve salient structures by matching keypoint coverage and local angles, maintaining informative locations and local corner geometry. The structural objective is $\mathcal{L}_{\text{structure}} = \lambda_{\text{spatial}}\mathcal{L}_{\text{spatial}} + \lambda_{\text{local}}\mathcal{L}_{\text{local}}$.

The full training loss combines teacher trajectory supervision, VCLT intermediate supervision, and the auxiliary terminal regularizer:

$$\mathcal{L} = \lambda_{\text{consistency}}\mathcal{L}_{\text{consistency}} + \lambda_{\text{VCLT}}\mathcal{L}_{\text{VCLT}} + \lambda_{\text{structure}}\mathcal{L}_{\text{structure}}. \tag{8}$$

Detailed formulations of the structural constraints are provided in Appendix C.

At inference time, we tile the partial scan P to the model point count and add Gaussian noise for initialization. The EMA model is then applied over $S+1$ evenly spaced teacher times: $Z_{n+1} = F_{\bar{\theta}}(Z_n, t_n, t_{n+1}; P)$. Here, S denotes the number of outer student transitions. Each transition internally executes the teacher-aligned VCLT micro-steps corresponding to the spanned teacher intervals, while the encoder and decoder are evaluated only once. A single LiTE checkpoint therefore supports different inference depths without retraining. For evaluation, the raw output may be passed through the fixed LiDiff refiner to obtain refined results; the refiner is outside student training.

3 Experiments

3.1 Experimental Setup

Data and training. We train LiTE on only 1,000 scans (5.2% of the full 19,130-scan SemanticKITTI training set) for 20 epochs on four H100 80 GB GPUs. We evaluate the resulting checkpoint on sequence 08 of SemanticKITTI and sequence 00 of KITTI-360 (Liao et al., 2023).

Baselines and metrics. We compare LMSCNet (Roldao et al., 2020), LODE (Li et al., 2023a), MID (Vizzo et al., 2022), PVD (Zhou et al., 2021), LiDiff (Nunes et al., 2024), LiDPM (Martyniuk et al., 2025), and ScoreLiDAR (Zhang et al., 2025). We report Chamfer distance (CD), bird's-eye-view Jensen–Shannon divergence (JSD), EMD using a deterministic 64-point approximation, and occupancy IoU at 0.5/0.2/0.1 m, accumulated across frames. Lower CD, JSD, and EMD and higher IoU indicate better completion.

3.2 Overall Results

Completion quality. We use four-transition LiTE as the main configuration in the experiments. In the comparison (Table 1), LiTE improves raw CD on SemanticKITTI by 11.6% over ScoreLiDAR and also improves JSD, EMD, and IoU at all three voxel sizes, so the gain is not confined to point proximity but extends to spatial support. Raw CD also falls below the LiDiff teacher on both

Table 1: Comparison on SemanticKITTI and KITTI-360. † marks refined outputs. Pale blue, pale pink, and cream cells indicate the best, second-best, and third-best value in each metric, respectively. "–" denotes an unavailable metric. LiTE uses **four steps** and ScoreLiDAR uses **eight**.

Method	SemanticKITTI						KITTI-360					
	CD↓	JSD↓	EMD↓	Voxel IoU↑			CD↓	JSD↓	EMD↓	Voxel IoU↑		
				0.5	0.2	0.1				0.5	0.2	0.1
LMSCNet (Roldao et al., 2020)	0.641	0.431	–	30.80	12.10	3.70	0.979	0.496	–	21.58	17.28	3.03
LODE (Li et al., 2023a)	1.029	0.451	–	33.80	16.40	5.00	1.565	0.483	–	–	–	–
MID (Vizzo et al., 2022)	0.503	0.470	–	31.60	22.70	13.10	0.637	0.476	–	28.15	20.20	11.91
PVD (Zhou et al., 2021)	1.256	0.498	–	15.90	4.00	0.60	–	–	–	–	–	–
LiDiff (Nunes et al., 2024)	0.434	0.444	22.15	31.50	16.80	4.70	0.564	0.459	21.98	29.35	16.65	4.86
LiDiff† (Nunes et al., 2024)	0.375	0.416	23.16	32.40	23.00	13.40	0.517	0.446	22.96	28.63	20.47	12.53
LiDPM (Martyniuk et al., 2025)	0.446	0.440	22.64	34.10	19.50	6.30	0.472	0.448	18.34	30.97	19.05	6.71
LiDPM† (Martyniuk et al., 2025)	0.377	0.403	23.47	36.60	25.80	14.90	0.415	0.423	18.24	31.53	22.61	14.11
ScoreLiDAR (Zhang et al., 2025)	0.406	0.425	23.14	33.65	13.11	3.26	0.472	0.444	22.80	28.14	13.45	4.08
ScoreLiDAR† (Zhang et al., 2025)	0.342	0.399	23.26	37.37	23.45	10.76	0.452	0.437	23.02	29.07	18.10	8.94
LiTE (ours)	0.359	0.395	18.71	34.82	17.19	4.93	0.447	0.407	17.55	31.03	12.91	3.29
LiTE (ours)†	0.283	0.361	18.74	36.82	25.81	13.67	0.377	0.388	17.69	31.93	20.92	9.18

datasets, indicating that compressing the trajectory does not sacrifice accuracy relative to the model it distills. Applying the shared refiner preserves this ordering, with refined LiTE reducing CD over refined ScoreLiDAR by 17.3% on SemanticKITTI and 16.6% on KITTI-360. The main exception is fine-scale IoU, where LiDPM remains higher; LiTE instead favors proximity and mid-scale support. Figure 4 illustrates the resulting surface recovery beyond scan lines while preserving the building corner and street layout.

Efficiency performance. We compare LiTE with ScoreLiDAR, the closest distillation-based competitor, at matched sampling depths (Figure 5(a)). LiTE has lower latency and CD than ScoreLiDAR at every matched depth; for example, at four steps it reaches CD 0.283 in 1.045 s, compared with 0.326 in 4.540 s for ScoreLiDAR. This advantage follows from LiTE's internal execution: ScoreLiDAR re-evaluates the full backbone at every step, whereas LiTE advances the corresponding teacher-interval span through VCLT inside one backbone pass. Full family models' step-wise measurements are reported in Appendix E.

3.3 Ablation Study

Structural supervision. Table 2 highlights the role of the transition formulation. Without terminal structural constraints, trajectory consistency already reaches 0.292 refined CD, already better

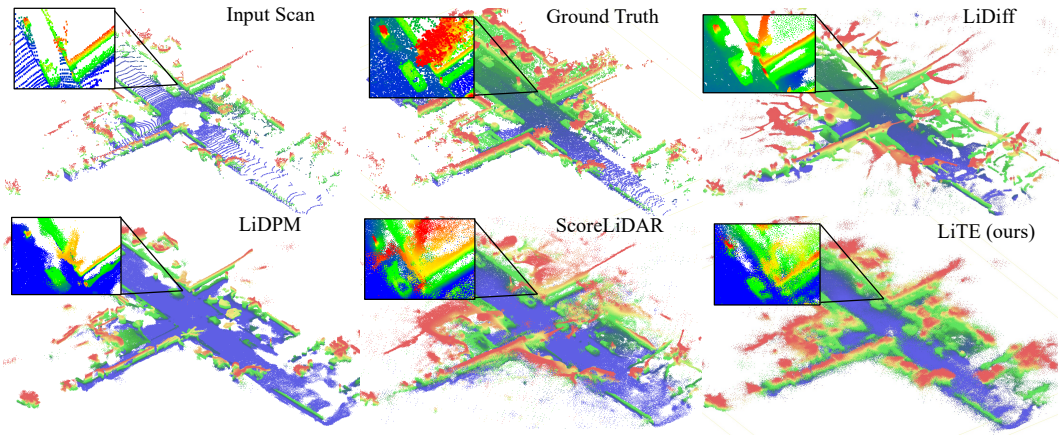


Figure 4: Qualitative completion comparison. Matched insets highlight building-corner geometry and surface continuity; the full views show the recovered street layout.

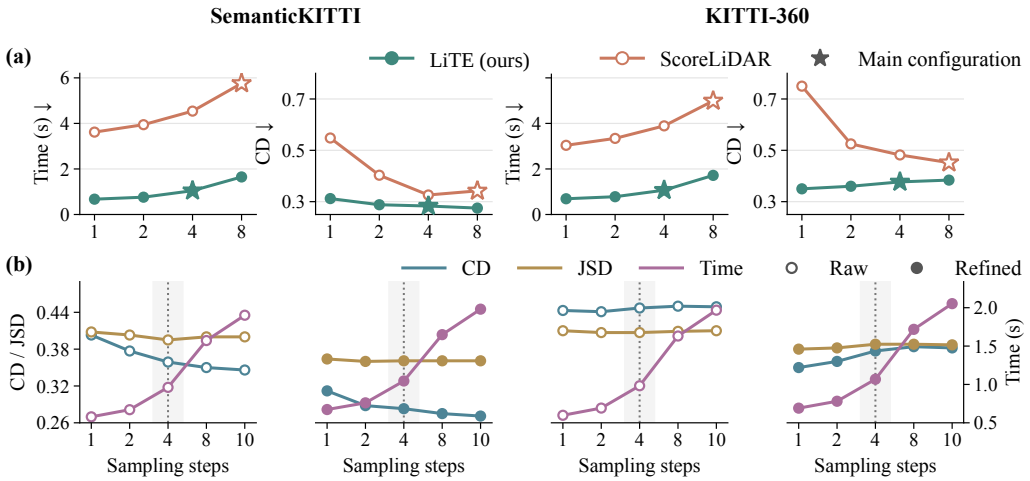


Figure 5: Sampling analysis. (a) Refined latency and CD at matched sampling depths. (b) LiTE sampling sweep; gray bands mark four transitions.

Table 2: Core transition-formulation ablations on SemanticKITTI at four steps. † denotes refined outputs. Pale blue, pale pink, and cream mark best, second-best, and third-best values.

Form.	Terminal	CD↓	CD† ↓	$\text{IoU}^{\dagger}_{0.5}$ ↑	$\text{IoU}^{\dagger}_{0.2}$ ↑	$\text{IoU}^{\dagger}_{0.1}$ ↑
Direct path	None	0.524	0.473	27.36	15.72	7.94
Direct path	Voxel + Geometry	0.434	0.385	28.08	16.21	8.31
Trajectory consistency	None	0.367	0.292	35.34	24.02	12.51
Trajectory consistency	Voxel + Geometry	0.359	0.283	36.82	25.81	13.67

than the direct path with both Voxel and Geometry constraints (0.385). With the same constraints, it further improves to 0.283, showing that the transition formulation contributes beyond terminal structural supervision. The direct path uses the $X + \Delta_\theta$ endpoint parameterization and does not include EMA endpoint agreement. This comparison therefore supports the transition formulation as a whole, rather than isolating EMA or internal micro-steps. Figure 6(a) then isolates the terminal structural constraints within trajectory consistency: voxel and geometry supervision provide complementary support and local-structure gains, with their combination giving the strongest result. Full results are in Appendix F.

Matching mechanism. The gain could come from a stronger matcher rather than visibility conditioning, so we compare VCLT with Closest Point and Radius Attention. Either replacement worsens

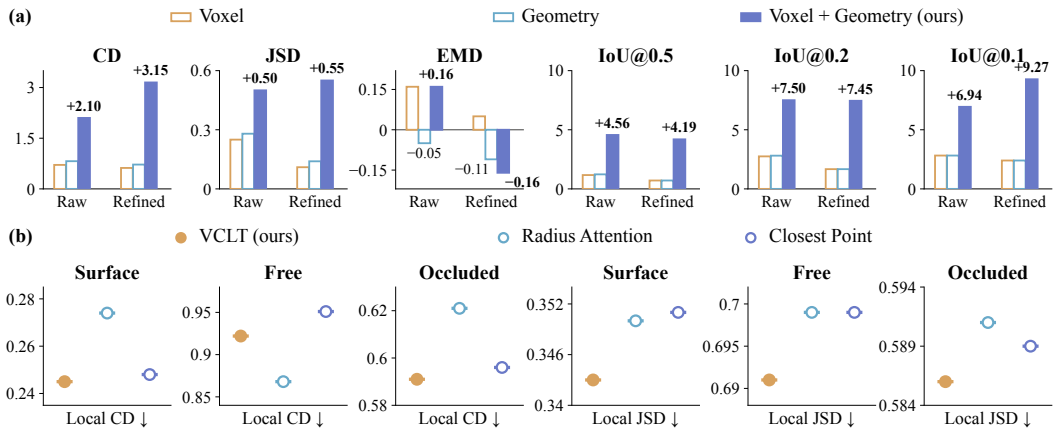


Figure 6: Component analysis. (a) Relative structural-constraint gains over the unconstrained trajectory model; full metrics are in Table 9. (b) Local errors by source ray class for Table 3(b).

refined CD and cuts 0.1 m IoU by more than two points (Table 3a). Breaking the error down by source ray class (Table 3b), VCLT attains the lowest local CD on surfaces and occluded regions and the lowest JSD in all classes, whereas Radius Attention is better only in free space (Figure 6b). The regional gains therefore match visibility-dependent conditioning rather than a uniformly stronger matcher.

Sampling depth and refinement. The sampling sweep in Figure 5(b) sets the main configuration. Raw SemanticKITTI CD improves up to four transitions and then saturates: extending to ten nearly doubles latency for only another 3.6% reduction. We therefore adopt four transitions. Refinement improves both point proximity and fine-scale occupancy, while its effect on EMD is comparatively small. It therefore complements, rather than merely duplicates, the effect of increasing the number of student transitions. Complete per-step measurements are in Appendix E.

Table 3: Matching ablations on SemanticKITTI at four steps. (a) Full-scene metrics; † denotes refinement. (b) Source-class local diagnostics.

Matching Type	CD↓	JSD↓	EMD↓	Voxel IoU↑		
				0.5	0.2	0.1
VCLT	0.359	0.395	18.71	34.82	17.19	4.93
VCLT†	0.283	0.361	18.74	36.82	25.81	13.67
Radius Attention	0.374	0.395	18.75	33.63	14.73	3.93
Radius Attention†	0.295	0.363	18.75	35.60	22.89	10.81
Closest Point	0.376	0.401	18.71	32.49	16.21	5.18
Closest Point†	0.311	0.363	18.71	34.43	22.17	11.28

(a) Overall matching-type ablation

Source Class	Matching Type	Local CD↓	Local JSD↓
Surface	VCLT	0.245	0.343
	Radius Attention	0.274	0.350
	Closest Point	0.248	0.351
Free	VCLT	0.922	0.691
	Radius Attention	0.868	0.699
	Closest Point	0.951	0.699
Occluded	VCLT	0.591	0.586
	Radius Attention	0.621	0.591
	Closest Point	0.596	0.589

(b) Source-class matching analysis

3.4 Detailed Components Cost

To characterize the cost of local conditioning, we separately profile the corresponding components in LiTE and ScoreLiDAR (Table 4). For LiTE, the teacher-aligned VCLT micro-step count is fixed across outer transition counts, so changing the count only repartitions the same workload and leaves the local-conditioning cost nearly constant. LiTE spends 0.331 s on VCLT transport, versus 0.990 s for ScoreLiDAR at its 8-step operating point, making LiTE about $3.0\times$ cheaper.

Table 4: Local-conditioning cost for LiTE and ScoreLiDAR.

Model	Configuration	Time (s)
LiTE	4 transitions (main configuration)	0.331
LiTE	8 transitions (matched to ScoreLiDAR)	0.339
ScoreLiDAR	8-step	0.990

4 Conclusion

Fast LiDAR scene completion requires preserving the geometric evolution of iterative generation without paying for a full backbone evaluation at every step. LiTE addresses this by executing each long student transition as an internal segment of the teacher trajectory: the backbone establishes the transition once, while a latent state carries its intermediate evolution and keeps local conditioning aligned as it moves.

Across SemanticKITTI and KITTI-360, LiTE delivers lower reconstruction error and substantially lower latency than the compared completion models. These results show that teacher-aligned internal evolution retains information that endpoint-only supervision loses, while allowing one model to operate at multiple inference depths. LiTE thus turns the trajectory structure of iterative completion into practical inference efficiency.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript editing and formatting, reference formatting and verification, and literature search. We have not used generative AI tools to develop the method, design or run the experiments, or produce the results, and these tasks are not applicable to this work. We have reviewed all AI-assisted work and verified every reference against its official record. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies LiDAR scene completion on the public SemanticKITTI and KITTI-360 driving datasets. It does not involve human subjects, personal data, or models that act in the physical world, and we do not foresee ethical concerns beyond those already associated with autonomous-driving perception research.

REPRODUCIBILITY STATEMENT

We specify the training budget and evaluation protocol in Section 3.1. The training objectives are defined in Sections 2.2–2.4 and detailed in Appendices C–D, the compared matching alternatives are specified in Appendix G, and the complete sampling and structural measurements are reported in Appendices E–F.

REFERENCES

Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9296–9306, 2019.

Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. AnchorFormer: Point Cloud Completion From Discriminative Nodes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13581–13590, 2023.

Ran Cheng, Christopher Agia, Yuan Ren, Xinhai Li, and Bingbing Liu. S3CNet: A Sparse Semantic Scene Completion Network for LiDAR Point Clouds. In *Proceedings of the 2020 Conference on Robot Learning*, pp. 2148–2161, 2021.

Yi Du, Zhipeng Zhao, Shaoshu Su, Sharath Golluri, Haoze Zheng, Runmao Yao, and Chen Wang. SuperPC: A Single Diffusion Model for Point Cloud Completion, Upsampling, Denoising, and Colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16953–16964, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, pp. 6840–6851, 2020.

Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013. 10.1007/s10514-012-9321-0.

Yi-Ting Hsiao, Siavash Khodadadeh, Kevin Duarte, Wei-An Lin, Hui Qu, Mingi Kwon, and Ratheesh Kalarot. Plug-and-Play Diffusion Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13743–13752, 2024.

Zixuan Huang, Justin Johnson, Shoubhik Debnath, James M. Rehg, and Chao-Yuan Wu. PointInfinity: Resolution-Invariant Point Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10050–10060, 2024.

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*, pp. 26565–26577, 2022.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

Pengfei Li, Ruowen Zhao, Yongliang Shi, Hao Zhao, Jirui Yuan, Guyue Zhou, and Ya-Qin Zhang. LODE: Locally Conditioned Eikonal Implicit Scene Completion from Sparse LiDAR. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8269–8276, 2023a.

Shanshan Li, Pan Gao, Xiaoyang Tan, and Mingqiang Wei. ProxyFormer: Proxy Alignment Assisted Point Cloud Completion With Missing Part Sensitive Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9466–9475, 2023b.

Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2023. 10.1109/TPAMI.2022.3179507.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo Numerical Methods for Diffusion Models on Manifolds. In *International Conference on Learning Representations (ICLR)*, 2022.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. InstaFlow: One Step is Enough for High-Quality Diffusion-Based Text-to-Image Generation. In *International Conference on Learning Representations (ICLR)*, 2024.

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*, pp. 5775–5787, 2022.

Shitong Luo and Wei Hu. Diffusion Probabilistic Models for 3D Point Cloud Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2836–2844, 2021.

Tetiana Martyniuk, Gilles Puy, Alexandre Boulch, Renaud Marlet, and Raoul de Charette. LiDPM: Rethinking Point Diffusion for Lidar Scene Completion. In *2025 IEEE Intelligent Vehicles Symposium (IV)*, pp. 555–560, 2025.

Lucas Nunes, Rodrigo Marcuzzi, Benedikt Mersch, Jens Behley, and Cyrill Stachniss. Scaling Diffusion Models to Real-World 3D LiDAR Scene Completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14770–14780, 2024.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*, 2017.

Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. LMSCNet: Lightweight Multiscale 3D Semantic Completion. In *2020 International Conference on 3D Vision (3DV)*, pp. 111–119, 2020.

Tim Salimans and Jonathan Ho. Progressive Distillation for Fast Sampling of Diffusion Models. In *International Conference on Learning Representations (ICLR)*, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations (ICLR)*, 2021a.

Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic Scene Completion From a Single Depth Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 190–198, 2017.

Yang Song and Prafulla Dhariwal. Improved Techniques for Training Consistency Models. In *International Conference on Learning Representations (ICLR)*, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations (ICLR)*, 2021b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32211–32252, 2023.

Junshu Tang, Zhijun Gong, Ran Yi, Yuan Xie, and Lizhuang Ma. LAKe-Net: Topology-Aware Point Cloud Completion by Localizing Aligned Keypoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1716–1725, 2022.

Ignacio Vizzo, Benedikt Mersch, Rodrigo Marcuzzi, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Make it Dense: Self-Supervised Geometric Scan Completion of Sparse 3D LiDAR Scans in Large Outdoor Environments. *IEEE Robotics and Automation Letters*, 7(3):8534–8541, 2022. 10.1109/LRA.2022.3187255.

Xiaogang Wang, Marcelo H. Ang, Jr., and Gim Hee Lee. Cascaded Refinement Network for Point Cloud Completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 787–796, 2020.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019. 10.1145/3326362.

Guangshun Wei, Yuan Feng, Long Ma, Chen Wang, Yuanfeng Zhou, and Changjian Li. PCDreamer: Point Cloud Completion Through Multi-view Diffusion Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27243–27253, 2025.

Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. PMP-Net: Point Cloud Completion by Learning Multi-Step Point Moving Paths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7439–7448, 2021.

Zhaoyang Xia, Youquan Liu, Xin Li, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, and Yu Qiao. SCPNet: Semantic Scene Completion on Point Cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17642–17651, 2023.

Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. SnowflakeNet: Point Cloud Completion by Snowflake Point Deconvolution With Skip-Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5479–5489, 2021.

Yuwen Xiong, Wei-Chiu Ma, Jingkang Wang, and Raquel Urtasun. Learning Compact Representations for LiDAR Completion and Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1074–1083, 2023.

Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse Single Sweep LiDAR Point Cloud Segmentation via Learning Contextual Shape Priors from Scene Completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3101–3109, 2021. 10.1609/aaai.v35i4.16419.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623, 2024.

Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PoinTr: Diverse Point Cloud Completion With Geometry-Aware Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12478–12487, 2021.

Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point Completion Network. In *2018 International Conference on 3D Vision (3DV)*, pp. 728–737, 2018.

Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. LION: Latent Point Diffusion Models for 3D Shape Generation. In *Advances in Neural Information Processing Systems*, pp. 10021–10039, 2022.

Shengyuan Zhang, An Zhao, Ling Yang, Zejian Li, Chenye Meng, Haoran Xu, Tianrun Chen, AnYang Wei, Perry Pengyun Gu, and Lingyun Sun. Distilling Diffusion Models to Efficient 3D LiDAR Scene Completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5007–5016, 2025.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16239–16248, 2021.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A Unified Predictor-Corrector Framework for Fast Sampling of Diffusion Models. In *Advances in Neural Information Processing Systems*, pp. 49842–49869, 2023.

Linqi Zhou, Yilun Du, and Jiajun Wu. 3D Shape Generation and Completion Through Point-Voxel Diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5806–5815, 2021.

APPENDIX OVERVIEW

This appendix provides supplementary material in six groups.

- **Extended related work.** Appendix A reviews related literature in more depth.
- **Training details.** Appendix B lists the training configuration used for the main LiTE model.
- **Method and objective details.** Appendix C gives the full endpoint and structural objectives, and Appendix D the intermediate path supervision.
- **Complete experimental results.** Appendix E reports all sampling and refinement measurements, and Appendix F the full structural ablations, including the direct-path comparison.
- **Baseline definitions.** Appendix G defines the matching alternatives used in the ablation study.
- **More visual results.** Appendix H shows additional qualitative completion examples.

A MORE RELATED WORKS

Scene and point-cloud completion. Semantic scene completion predicts the occupied and semantic structure of a scene from partial observations. SSCNet established the task by jointly predicting occupancy and semantics from a single depth image (Song et al., 2017), and SemanticKITTI and KITTI-360 extended evaluation to outdoor LiDAR sequences (Behley et al., 2019; Liao et al., 2023). Early LiDAR methods represent the scene as a voxel grid and apply multiscale or sparse convolutions: LMSCNet performs lightweight multiscale completion (Roldao et al., 2020), S3CNet operates directly on sparse point clouds (Cheng et al., 2021), and SCPNet couples completion with segmentation and uses multiple input sweeps (Xia et al., 2023); a related line uses completion as a shape prior for segmentation (Yan et al., 2021). Because the grid is memory-bound, later work replaces explicit voxels with continuous fields, including self-supervised signed-distance fields (Vizzo et al., 2022), locally implicit eikonal surfaces (Li et al., 2023a), and compact discrete representations (Xiong et al., 2023). At the object level, PCN introduced a learned point decoder (Yuan et al., 2018), CRN refines coarse predictions across scales (Wang et al., 2020), and SnowflakeNet generates points through snowflake deconvolution (Xiang et al., 2021); PMP-Net learns multi-step point moving paths (Wen et al., 2021), PoinTr casts completion as a set-to-set transformer problem (Yu et al., 2021), AnchorFormer learns discriminative anchor nodes (Chen et al., 2023), ProxyFormer aligns proxy features for the missing part (Li et al., 2023b), and LAKe-Net localizes aligned keypoints to preserve topology (Tang et al., 2022). These methods show that generating points through a sequence of displacements is more powerful than a single regression, which motivates our trajectory view. They are, however, either bound to a grid or optimization pipeline, or object-centric, and none of them addresses the cost of iteratively re-evaluating the generative model or the reliability of scan conditioning at scene scale.

Diffusion-based geometry and conditioning. Diffusion models provide the generative backbone of our method. Point diffusion generates shapes by denoising point sets (Luo & Hu, 2021), point–voxel diffusion combines explicit points with a voxel backbone (Zhou et al., 2021), LION operates in a latent space (Zeng et al., 2022), and PointInfinity makes point diffusion resolution-invariant (Huang et al., 2024); SuperPC and PCDreamer extend point diffusion to restoration, upsampling, and conditional completion with image priors (Du et al., 2025; Wei et al., 2025). For outdoor scenes, LiDiff applies DDPMs directly to scene-scale point clouds (Nunes et al., 2024) and LiDPM reformulates the noise schedule for point-level diffusion (Martyniuk et al., 2025). These models produce detailed geometry but recover a scene through many denoising steps. Conditioning them on the observed scan requires aggregating local evidence: PointNet++ provides hierarchical set abstraction (Qi et al., 2017) and the Point Transformer introduces local attention (Zhao et al., 2021), while DGCNN builds dynamically updated neighborhood graphs (Wang et al., 2019); ray-based occupancy mapping further separates measured surfaces, free space, and unknown regions (Hornung et al., 2013). We combine these ideas: VCLT re-matches queries as they move and routes local versus global motion according to sensor-ray visibility, which fixed nearest-neighbor conditioning cannot do.

Efficient sampling and distillation. DDPMs and score-based models connect generation to iterative denoising and probability-flow integration (Ho et al., 2020; Song et al., 2021b). One route to acceleration uses better numerical solvers: DDIM takes deterministic implicit steps (Song et al., 2021a), PNDM and DPM-Solver raise the solver order (Liu et al., 2022; Lu et al., 2022), and UniPC unifies predictor–corrector sampling (Zhao et al., 2023); another redesigns the noise schedule and preconditioning (Karras et al., 2022). These methods reduce the number of evaluations but still evaluate the full denoiser at every retained step, so their cost grows with the number of steps. Distillation instead moves the acceleration cost into training. Progressive distillation halves the number of steps by matching a 2-step teacher update (Salimans & Ho, 2022); consistency models learn a direct mapping from any noise level to the endpoint (Song et al., 2023; Song & Dhariwal, 2024); and CTM generalizes consistency to arbitrary trajectory intervals (Kim et al., 2024). Distribution matching and plug-and-play distillation accelerate image generators through distributional or adversarial objectives (Yin et al., 2024; Hsiao et al., 2024), and flow matching and rectification construct near-straight probability paths for 1-step generation (Lipman et al., 2023; Liu et al., 2024). For LiDAR completion, ScoreLiDAR distills LiDiff into an 8-step student and adds structural supervision (Zhang et al., 2025). These methods shorten the path but still propagate each transition through the entire student network. Building on CTM's arbitrary-interval transition parameterization, LiTE focuses on internal execution: teacher-aligned latent states evolve inside one student evaluation, moving queries refresh visibility-dependent conditioning, and terminal structural supervision constrains the final output.

B Training Details

We use the same configuration for the main SemanticKITTI experiments and the matching ablations unless an ablation explicitly changes the model variant. Table 5 records the settings that affect training; evaluation-only settings and filesystem paths are omitted. The values are taken from the resolved configuration of the official runs.

Table 5: Training configuration for LiTE.

Group	**Setting**
Data	SemanticKITTI sequences 00–07, 09, 10 for training and sequence 08 for validation; 100 training frames and 20 validation frames per sequence.
Representation	Voxel resolution $0.05\,\mathrm{m}$; 180,000 points per scan; range clipped to $50\,\mathrm{m}$.
Teacher	LiDiff teacher with 50 denoising steps and the linear schedule used by LiDiff ($\beta_{\mathrm{start}} = 3.5 \times 10^{-5}$, $\beta_{\mathrm{end}} = 0.007$).
Sampling	EMA-anchor training; $\Delta t_{\max} = 50$ with weighted interval-span sampling; uniformly sampled start time; EMA-anchor probability 0.8.
Student	Recurrent trajectory-consistency transition with $d_{\mathrm{out}} = 96$; four attention heads of dimension 128 at stage $x4$; 8 coarse and 4 fine candidates; EMA target rate 0.99.
Objective	$\mathcal{L}_{\mathrm{consistency}}$ with $\mathcal{S}_{\boldsymbol{\lambda}}$ point-set terms and Huber $\beta = 0.1$; $\mathcal{L}_{\mathrm{VCLT}}$ combines position, direction, and free-space supervision with weight 0.5.
Structural terms	Voxel sizes $(0.1, 0.2)\,\mathrm{m}$ with weights $(0.7, 0.3)$; precision, recall, and detail weights $(1.0, 0.05, 1.0)$.
Optimizer	Adam-style training with learning rate 10^{-4}, zero weight decay, gradient clipping at 10, and a half-cosine schedule starting after 40% of training.
Runtime	Four H100 80 GB GPUs; batch size 2 per process; two data-loader workers per process; 32-bit precision; 20 epochs; random seed 42.
Selection	Validation every epoch; checkpoints ranked by validation detail score, with the best five retained.

The four-transition model used in the main comparisons is obtained by applying the same trained transition operator four times at inference. The ablation models retain the data, sampler, optimizer, and loss settings above and change only the component under study.

C Detailed Endpoint and Structural Objectives

C.1 Direct teacher supervision

The EMA endpoint target changes as training proceeds and can lag the online model, so we anchor each transition to the recorded teacher state as well. For point n, let $\widehat{d}_n = \widehat{x}_{j,n} - x_{i,n}$ and $d_n^{\mathrm{T}} = x_{j,n} - x_{i,n}$ denote the predicted and teacher displacements. We define $\mathcal{M} = \{n : \|d_n^{\mathrm{T}}\|_2 > \epsilon\}$ and $\boldsymbol{\lambda}^{\mathrm{T}} = (\lambda_{\mathrm{ta}}, \lambda_{\mathrm{tp}}, \lambda_{\mathrm{tc}})$. The direct teacher loss is

$$\begin{aligned}\mathcal{L}_{\mathrm{teacher}} = \mathcal{S}_{\boldsymbol{\lambda}^{\mathrm{T}}}(\widehat{X}_j, X_j) \\ + \frac{\lambda_{\mathrm{td}}}{|\mathcal{M}|} \sum_{n\in\mathcal{M}} \Big[1 - \cos(\widehat{d}_n, d_n^{\mathrm{T}})\Big] \\ + \frac{\lambda_{\mathrm{tm}}}{|\mathcal{M}|} \sum_{n\in\mathcal{M}} \ell_{\mathrm{H}}\left(\log(1+\|\widehat{d}_n\|_2), \log(1+\|d_n^{\mathrm{T}}\|_2)\right). \end{aligned} \tag{9}$$

C.2 Structural supervision

Matching point positions constrains proximity but not where points are distributed or whether local shape is preserved. The structural terms therefore compare the terminal prediction Z with the complete ground truth Y at two levels, and the following components constitute the structural objective.

Multiscale occupancy and density. Point level matching does not prevent many predicted points from concentrating in the same region. We therefore compare how Z and Y occupy space. To retain coordinate gradients, each point distributes trilinear mass m_v to its eight neighboring voxels, which is converted to a saturating occupancy $o_v = 1 - \exp(-m_v)$. For voxel sizes $h \in \{0.1, 0.2\}$ m, we use

$$\mathcal{L}^h_{\mathrm{occupancy}} = 1 - \frac{2\langle o^Z, o^Y\rangle + \epsilon}{\|o^Z\|_2^2 + \|o^Y\|_2^2 + \epsilon}, \qquad \mathcal{L}^h_{\mathrm{density}} = \ell_{\mathrm{H}}\big(\log(1+\bar{m}^Z), \log(1+\bar{m}^Y)\big) \tag{10}$$

where $\bar{m}$ denotes mass normalized by the mean positive mass at that scale. The Dice term aligns fine and coarse occupied support. Because occupancy saturates once a voxel is filled, it cannot detect duplicate points inside that voxel. The normalized density term restores this information and penalizes many to one point collisions.

Salient geometry preservation. Voxel occupancy records where points are present, but not whether a local corner retains its shape. We therefore extract 512 GT keypoints using dual scale ISS/PCA saliency. The ratio of the smallest to largest neighborhood covariance eigenvalue is high where geometry varies along all three axes and therefore favors corner structures. For keypoint p_k with saliency s_k, let $\widehat{p}_k = \mathrm{NN}_Z(p_k)$ be its predicted representative. We cover the salient structures and preserve the angles of selected local and structural triplets through

$$\mathcal{L}_{\mathrm{keypoint}} = \frac{\sum_k s_k \ell_{\mathrm{H}}(\widehat{p}_k, p_k)}{\sum_k s_k + \epsilon}, \qquad \mathcal{L}_{\mathrm{angle}} = \ell_{\mathrm{H}}(\cos\widehat{\phi}, \cos\phi). \tag{11}$$

Here, ϕ is the GT angle of a keypoint triplet and $\widehat{\phi}$ is the angle formed by their predicted representatives. Keypoint coverage retains informative locations, while angle matching prevents corners from being flattened without requiring dense geometric labels.

D Detailed Intermediate Path Supervision

The recurrent matcher operates on sparse query locations, whereas the teacher trajectory retains the identities of all particles. Let $q_m^{(\ell)}$ be query m after micro step ℓ, and let $x_{i,n}$ be source particle n.

We assign each source particle to its nearest initial query,

$$\pi(n) = \arg\min_m \|x_{i,n} - q_m^{(0)}\|_2, \qquad \widehat{x}_n^{(\ell)} = x_{i,n} + q_{\pi(n)}^{(\ell)} - q_{\pi(n)}^{(0)}. \tag{12}$$

This transfer produces a full predicted cloud $\widehat{X}^{(\ell)}$ at every micro step. Its target is the LiDiff state $X_{i+\ell}$ at the same teacher interval. We average their smooth L_1 distance along the path,

$$\mathcal{L}_{\text{position}} = \frac{1}{L} \sum_{\ell=1}^{L} \ell_{\text{H}}\big(\widehat{X}^{(\ell)}, X_{i+\ell}\big). \tag{13}$$

Position matching alone does not specify how particles move between adjacent states. Define the predicted and teacher interval displacements as

$$\Delta\widehat{x}_n^{(\ell)} = \widehat{x}_n^{(\ell)} - \widehat{x}_n^{(\ell-1)}, \qquad \Delta x_n^{\text{T},(\ell)} = x_{i+\ell,n} - x_{i+\ell-1,n}. \tag{14}$$

Let $\mathcal{M}_\ell$ contain particles whose teacher displacement is larger than a small threshold. Their direction loss is

$$\mathcal{L}_{\text{direction}} = \frac{1}{\sum_\ell |\mathcal{M}_\ell|} \sum_{\ell=1}^{L} \sum_{n \in \mathcal{M}_\ell} \left[1 - \cos\big(\Delta\widehat{x}_n^{(\ell)}, \Delta x_n^{\text{T},(\ell)}\big)\right]. \tag{15}$$

Finally, a point routed to free space should not move toward the sensor. Let $w_{n,\text{free}}^{(\ell)}$ be its lifted free space weight and $e_n^{(\ell-1)} = \widehat{x}_n^{(\ell-1)} / \|\widehat{x}_n^{(\ell-1)}\|_2$ its outward radial direction. We use

$$\mathcal{L}_{\text{freespace}} = \frac{\sum_{\ell,n} w_{n,\text{free}}^{(\ell)} \left[-\Delta\widehat{x}_n^{(\ell)\top} e_n^{(\ell-1)}\right]_+}{\sum_{\ell,n} w_{n,\text{free}}^{(\ell)} + \epsilon}. \tag{16}$$

Equations 13, 15, and 16 provide the detailed terms used to instantiate the VCLT objective in the main text.

E Complete Sampling and Refinement Results

This appendix reports the complete sampling and refinement measurements that support the main text, keeping raw and refined outputs separate. Unless stated otherwise, all inference times use the same four H100 80 GB GPUs and runtime conditions as training; baseline methods are rerun under this common hardware and measurement protocol to control the comparison.

E.1 Matched comparison with ScoreLiDAR

ScoreLiDAR is the closest distillation-based competitor, so we compare the two over a range of matched sampling depths rather than at a single operating point. As Figure 5(a) shows, LiTE has lower latency and CD at every matched depth on both datasets. On SemanticKITTI, for example, four-step LiTE reaches CD 0.283 in 1.045 s, compared with 0.326 in 4.540 s for four-step ScoreLiDAR; the eight-step comparison is likewise 0.275 versus 0.342 at 1.649 versus 5.757 s. ScoreLiDAR's CD worsens when moving from four to eight steps on SemanticKITTI, while it improves with depth on KITTI-360. These measurements reflect the computation unit: each ScoreLiDAR step re-evaluates the full backbone, whereas LiTE advances the corresponding teacher-interval span through VCLT inside one backbone pass. Table 6 provides all measurements for this comparison, retaining CD, JSD, EMD, and latency at each sampling depth.

Table 6: Sampling-depth comparison with refinement between LiTE (ours) and ScoreLiDAR on SemanticKITTI and KITTI-360. Each dataset reports CD, JSD-BEV, EMD, and measured inference time including refinement. Each dataset uses its stated evaluation split. Colors are assigned by global ranking over all rows within each dataset and metric, with pale blue, pale pink, and cream denoting the best, second-best, and third-best values. "–" denotes an unavailable metric.

		SemanticKITTI				KITTI-360			
Model	Steps	CD↓	JSD↓	EMD↓	Time (s)↓	CD↓	JSD↓	EMD↓	Time (s)↓
LiTE (ours)	1	0.312	0.364	18.83	0.673	0.350	0.380	17.92	0.691
	2	0.288	0.360	18.77	0.760	0.360	0.382	17.59	0.781
	4	0.283	0.361	18.74	1.045	0.377	0.388	17.69	1.068
	8	0.275	0.361	18.80	1.649	0.384	0.388	17.95	1.718
ScoreLiDAR (Zhang et al., 2025)	1	0.548	0.384	–	3.618	0.750	0.478	–	3.038
	2	0.403	0.379	–	3.944	0.525	0.457	–	3.342
	4	0.326	0.386	23.98	4.540	0.482	0.461	23.76	3.892
	8	0.342	0.399	23.26	5.757	0.452	0.437	23.02	4.989

Across the matched rows, LiTE consistently combines lower latency with lower CD than ScoreLiDAR on both datasets. The advantage persists as the sampling depth increases, while ScoreLiDAR pays for each additional step with another full-backbone evaluation.

E.2 Effect of sampling depth and refinement

The sampling sweep in Figure 5(b) sets the main configuration. Raw SemanticKITTI CD improves up to four transitions and then saturates: extending to ten nearly doubles latency for only another 3.6% reduction, and JSD is already best at four. KITTI-360 raw quality changes little with depth. We therefore adopt four transitions, which captures the trajectory-level gain before the tradeoff turns unfavorable. Refinement improves both point proximity and fine-scale occupancy, while its effect on EMD is comparatively small. It therefore complements, rather than merely duplicates, the effect of increasing the number of student transitions. All step counts on both datasets use this same LiTE checkpoint and evaluation protocol, so this separates changes due to sampling depth from retraining or model selection; the raw and refined blocks also expose the contribution of the reused refinement network.

Table 7: Ablation of sampling steps and refinement for LiTE (ours) on SemanticKITTI and KITTI-360. Each dataset reports CD, JSD-BEV, EMD, voxel IoU at sizes 0.5, 0.2, and 0.1 m, and inference time in seconds (including refinement where enabled). Colors are assigned by global ranking over all ten rows within each dataset and metric; pale blue, pale pink, and cream denote the best, second-best, and third-best values, respectively.

		SemanticKITTI							KITTI-360						
Refinement	Steps	CD↓	JSD↓	EMD↓	Voxel IoU↑			Time↓	CD↓	JSD↓	EMD↓	Voxel IoU↑			Time↓
					0.5	0.2	0.1					0.5	0.2	0.1	
	1	0.403	0.408	18.82	33.41	14.46	3.64	0.578	0.443	0.410	17.85	31.76	12.14	2.79	0.598
	2	0.377	0.403	18.78	35.19	16.64	4.54	0.671	0.441	0.407	17.61	32.02	13.10	3.18	0.692
	4	0.359	0.395	18.71	34.82	17.19	4.93	0.962	0.447	0.407	17.55	31.03	12.91	3.29	0.983
	8	0.350	0.400	18.76	33.92	17.18	5.19	1.572	0.450	0.409	17.85	30.66	13.22	3.41	1.631
	10	0.346	0.400	18.75	33.79	17.02	5.06	1.902	0.449	0.410	17.90	30.93	13.16	3.26	1.968
✓	1	0.312	0.364	18.83	33.75	22.82	10.04	0.673	0.350	0.380	17.92	32.83	21.10	8.10	0.691
✓	2	0.288	0.360	18.77	37.93	26.89	13.21	0.760	0.360	0.382	17.59	33.39	22.36	9.35	0.781
✓	4	0.283	0.361	18.74	36.82	25.81	13.67	1.045	0.377	0.388	17.69	31.93	20.92	9.18	1.068
✓	8	0.275	0.361	18.80	36.43	25.00	13.54	1.649	0.384	0.388	17.95	31.28	20.38	9.34	1.718
✓	10	0.271	0.361	18.72	36.33	24.91	13.43	1.981	0.382	0.387	17.96	31.60	20.41	9.22	2.053

The complete sweep confirms that refinement supplies a substantial occupancy gain at every depth, whereas increasing the number of transitions mainly improves CD until the four-transition regime and then saturates. This is the quality–cost trade-off used for the main configuration.

Table 8 reports the quality and latency of the main configurations, and Figure 5(b) visualizes the wider sampling sweep.

Table 8: Quality and inference latency on SemanticKITTI. JSD denotes JSD-BEV. † marks refined outputs. LiDiff, LiDPM, ScoreLiDAR, and LiTE use 50, 20, 8, and 4 sampling steps, respectively. Time is measured in seconds; refined rows include the refiner pass. Within each block, pale blue and pale pink cells indicate the best and second-best value in each metric, respectively.

Method	CD↓	JSD↓	EMD↓	Voxel IoU↑			Time↓
				0.5	0.2	0.1	
LiDiff (Nunes et al., 2024)	0.434	0.444	22.15	31.50	16.80	4.70	17.710
LiDPM (Martyniuk et al., 2025)	0.446	0.440	22.64	34.10	19.50	6.30	12.353
ScoreLiDAR (Zhang et al., 2025)	0.406	0.425	23.14	33.65	13.11	3.26	5.670
LiTE (ours)	0.359	0.395	18.71	34.82	17.19	4.93	0.962
LiDiff† (Nunes et al., 2024)	0.375	0.416	23.16	32.40	23.00	13.40	17.797
LiDPM† (Martyniuk et al., 2025)	0.377	0.403	23.47	36.60	25.80	14.90	12.453
ScoreLiDAR† (Zhang et al., 2025)	0.342	0.399	23.26	37.37	23.45	10.76	5.757
LiTE (ours)†	0.283	0.361	18.74	36.82	25.81	13.67	1.045

The main comparison shows that LiTE has the best raw and refined CD, EMD, and latency among the listed methods, while retaining competitive voxel occupancy. The refined block preserves this ordering and adds the largest gain at the finer voxel scales.

E.3 Isolated conditioning cost

To characterize the cost of local conditioning, we separately profile the corresponding components in LiTE and ScoreLiDAR; the measurements are summarized in Table 4. Let S be the number of outer transitions and L_s the number of teacher intervals assigned to transition s. The full LiTE rollout cost decomposes as

$$T_{\mathrm{LiTE}}(S) \approx S\, C_{\mathrm{backbone}} + \left(\sum_{s=1}^{S} L_s\right) C_{\mathrm{micro}} + C_{\mathrm{fixed}}, \qquad \sum_{s=1}^{S} L_s = M, \tag{17}$$

where C_{backbone} is one encoder-decoder evaluation, C_{micro} is one VCLT micro-step, and C_{fixed} collects setup overhead. The fixed interval budget M means that changing S only repartitions the same teacher-aligned VCLT micro-steps; the table profiles the local-conditioning component rather than attributing the full rollout difference to architecture alone. The small difference between the two LiTE measurements comes from per-transition overhead, while ScoreLiDAR repeats its conditioning branches at every sampling step.

F Complete Structural Ablation Results

The main text isolates the structural supervision on the trajectory-consistency formulation, but the completion gain could come either from that formulation or from the terminal structural supervision. To separate the two, Table 9 reports all six metrics for both the trajectory-consistency and direct-path formulations, each evaluated with no additional structural constraints, voxel constraints, geometry constraints, or both, at four sampling steps, with raw and refined results reported separately.

On the trajectory-consistency formulation, voxel or geometry constraints alone improve only marginally over no constraints, as raw CD stays at 0.364 against 0.367 and refined CD at 0.290 against 0.292. Adding both produces a clear gain, lowering raw CD to 0.359 and refined CD to 0.283 while raising 0.2 m IoU from 24.02 to 25.81. The two families therefore cover complementary aspects of the terminal geometry, and their combination gives the strongest CD and occupancy. The larger relative occupancy gains than CD gains in Figure 6(a) are consistent with this supervision targeting spatial support and local geometry.

We add a direct-path comparison formulation that predicts the destination displacement directly, $X + \Delta_\theta(X, t, s; P)$, and is trained by direct teacher endpoint supervision, without the EMA endpoint agreement across starting states. Both formulations share the same network, the same four transitions, and the same four constraint settings, so the transition objective is the controlled variable. Structural supervision helps in both cases, so its benefit is not tied to the transition formulation: with voxel and geometry constraints, refined CD falls from 0.292 to 0.283 for trajectory consistency and from 0.473 to 0.385 for the direct path. The two formulations nevertheless remain far apart, with trajectory consistency leading by 0.385 versus 0.283 refined CD and 9.60 points of 0.2 m IoU once both constraints are active. The transition formulation is therefore the main source of completion quality, and structural supervision is complementary rather than a substitute. The direct path also depends on the two constraint families together: voxel or geometry alone leaves its raw CD nearly unchanged (0.524 to 0.512 and 0.522), whereas combining them lowers it to 0.434.

Figure 6(a) uses the unrounded measurements to compute relative changes, with each output stage compared against its own baseline without either constraint family.

Table 9: Complete structural ablations on SemanticKITTI at four steps. Pale blue, pale pink, and cream mark the three best values per metric across all 16 outputs; ties share a color. The trajectory-consistency block matches Table 2.

Constraints	Output	CD↓	JSD↓	EMD↓	$\text{IoU}_{0.5}$ ↑	$\text{IoU}_{0.2}$ ↑	$\text{IoU}_{0.1}$ ↑
Trajectory consistency							
None	Raw	0.367	0.397	18.74	33.30	15.99	4.61
	Refined	0.292	0.363	18.71	35.34	24.02	12.51
Voxel	Raw	0.364	0.396	18.71	33.69	16.43	4.74
	Refined	0.290	0.362	18.70	35.59	24.42	12.81
Geometry	Raw	0.364	0.396	18.75	33.71	16.44	4.74
	Refined	0.290	0.362	18.73	35.59	24.42	12.81
Voxel + Geometry	Raw	0.359	0.395	18.71	34.82	17.19	4.93
	Refined	0.283	0.361	18.74	36.82	25.81	13.67
Direct path							
None	Raw	0.524	0.396	18.95	27.54	13.43	4.10
	Refined	0.473	0.374	18.98	27.36	15.72	7.94
Voxel	Raw	0.512	0.395	18.97	27.92	13.74	4.13
	Refined	0.460	0.373	18.97	27.57	16.07	8.18
Geometry	Raw	0.522	0.395	19.00	27.93	13.73	4.13
	Refined	0.470	0.373	18.98	27.61	16.09	8.18
Voxel + Geometry	Raw	0.434	0.390	18.88	28.11	13.99	4.34
	Refined	0.385	0.366	18.88	28.08	16.21	8.31

G Matching Baselines

To make the matching comparison reproducible, this appendix defines the two conditioning alternatives evaluated in Table 3 and Figure 6(b). Both use a sparse U-Net to predict the student transition, with separate embeddings of the starting and destination times, so the comparison concerns how partial-scan features condition this prediction. Neither alternative performs the micro-step query-coordinate updates or sensor-ray routing of VCLT.

G.1 Closest Point Matching

We first ask whether a single nearest observation is sufficient conditioning for the distilled transition. Let q_i be a query location at a U-Net stage and let $\{(p_j, f_j)\}$ denote the encoded partial-scan locations and features. Closest Point selects the spatially nearest encoded observation,

$$j^*(i) = \arg\min_j \|q_i - p_j\|_2^2, \qquad c_i = f_{j^*(i)}. \tag{18}$$

The selected feature is transformed by a learned multilayer perceptron and concatenated with the two time embeddings. A second perceptron maps this concatenation to a channel-wise modulation vector, which multiplies the query features before the next encoder or decoder stage. Thus, conditioning depends on both the observed geometry and the requested transition interval.

Matching is evaluated at the sparse query locations of each stage. It is not a single index shared across all U-Net resolutions, nor does it update the query coordinates through intermediate teacher states. Each lookup returns one feature, without attention over multiple observations or ray-class selection.

G.2 Radius Attention Matching

We next test whether aggregating a broader neighborhood, rather than routing a single match by visibility, is sufficient. Radius Attention replaces nearest-observation conditioning with a residual cross-attention module at the deepest encoder stage. Queries are obtained from the current query feature concatenated with the starting and destination time embeddings; keys and values are linear projections of partial-scan features at the corresponding encoder scale. The module uses four attention heads and a total projection dimension of 128, giving 32 channels per head. The nearest-point conditioning branches are disabled in this baseline.

For query q_i, first retrieve up to 64 nearest partial-scan locations. Retain those within an interval-span-dependent radius $r(g)$, where g is the number of teacher intervals spanned by the transition and hence its VCLT micro-steps. To avoid an empty or very small neighborhood, always retain the eight nearest candidates, or all available candidates when fewer than eight exist. Consequently, this fallback can admit points outside the nominal radius. Writing the resulting candidate set as $\mathcal{N}_i(g)$, a single attention head computes

$$\alpha_{ij} = \frac{\exp(Q_i^\top K_j/\sqrt{32})}{\sum_{k\in\mathcal{N}_i(g)} \exp(Q_i^\top K_k/\sqrt{32})}, \qquad a_i = \sum_{j\in\mathcal{N}_i(g)} \alpha_{ij} V_j. \tag{19}$$

The four head outputs are concatenated, projected to the query-feature dimension, and added residually. The attention weights aggregate features; they do not directly displace query coordinates.

The radius is calibrated from teacher-particle displacement statistics. At each tabulated interval span, it equals the 95th percentile of displacement plus $0.5\,\mathrm{m}$; intermediate spans use linear interpolation. The calibration uses spans $1, 3, 5, 12, 13, 25, 50$. For example, the radii are approximately $0.730\,\mathrm{m}$ at span 1, $3.525\,\mathrm{m}$ at span 13, and $10.833\,\mathrm{m}$ at span 50. Distances for candidate selection are expressed in metric coordinates. The optional distance-based visibility gate is disabled in the evaluated configuration, and no sensor-ray classes enter this attention operation.

G.3 Full matching and visibility results

VCLT conditions the transition on deep features that are re-matched as the trajectory evolves, so a natural concern is that any deeper matcher, rather than visibility conditioning in particular, could explain the gain. We therefore compare against two alternatives: Closest Point retrieves the nearest scan feature at each U-Net stage, and Radius Attention aggregates an interval-span-dependent neighborhood at the deepest stage. Either replacement worsens raw and refined CD (Table 3a), with a refined 0.1 m IoU deficit exceeding two points. Nearly identical raw JSD for VCLT and Radius Attention suggests that the benefit concerns local support. This comparison tests VCLT as a whole, not its transport and routing components independently.

Aggregate metrics cannot show whether visibility routing helps where it is meant to, namely near surfaces and behind occluded returns, so we break the local error down by source ray class using source-assigned sampled ground truth (Table 3b). VCLT gives the lowest local CD on surfaces and occluded regions and the lowest JSD in all classes (Figure 6b), whereas Radius Attention has lower free-space CD. The regional gains near surfaces and behind returns are therefore consistent with visibility-dependent conditioning rather than with a uniformly stronger matcher.

Scope of the comparison. Closest Point provides a single spatial correspondence at each conditioning stage; Radius Attention aggregates a broader, interval-span-dependent neighborhood at the deepest encoder stage. VCLT instead refreshes correspondences as queries move and routes their influence using sensor-ray visibility. Table 3(a) reports the resulting full-scene quality before and after refinement, while Table 3(b) uses the same source-class diagnostic for all three alternatives. These are comparisons of complete conditioning designs, including their placement and aggregation rules; they do not isolate the independent effects of latent transport and routing.

H MORE VISUAL RESULTS

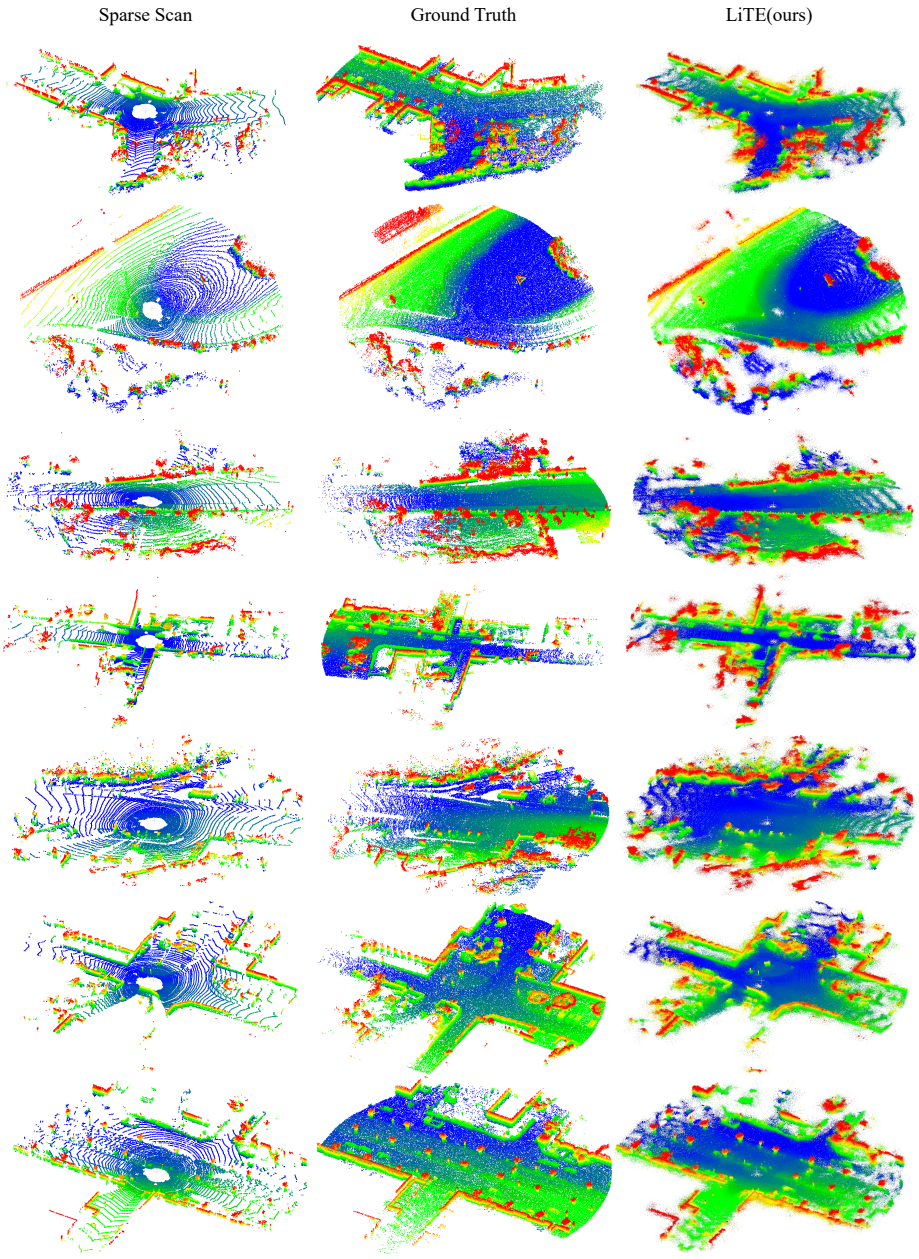


Figure 7: Additional qualitative completion results. Each row compares a sparse scan, the complete ground truth, and the LiTE completion.

Figure 7 shows additional qualitative completions from SemanticKITTI. We select one frame from each of sequences 00, 01, 02, 03, 04, 05, and 06, giving seven scenes that cover different road layouts, building arrangements, vegetation patterns, and degrees of scan sparsity. These are held-out frames: although their sequence IDs overlap with the training sequences, none of the selected frames is included in the training subset. Each row corresponds to one selected frame, while the three columns show the same scene as the sparse input scan, the complete ground truth, and the LiTE prediction, respectively. The input scans retain the characteristic scan-line structure and leave substantial regions unobserved; the ground-truth columns show the target scene support used for comparison.

Across these examples, LiTE extends the observed surfaces into the missing parts of the scene while preserving the dominant road and building geometry. The predictions recover broad planar regions, roadside structures, and the overall spatial arrangement of the visible scene. Fine details remain more variable in heavily occluded or weakly observed regions, but the completions retain coherent large-scale structure rather than simply copying the visible scan.

I Code and Model Release

The anonymous implementation and released model checkpoint are available at the following URL:

`https://anonymous.4open.science/r/LiTE-LiDAR-Trajectory-Evolution-27FC/`