UNICOMPLETE: UNIFIED MULTIMODAL HIGH-RESOLUTION 3D SHAPE COMPLETION VIA ANCHORED LATENT SET

Anonymous authors
Paper under double-blind review

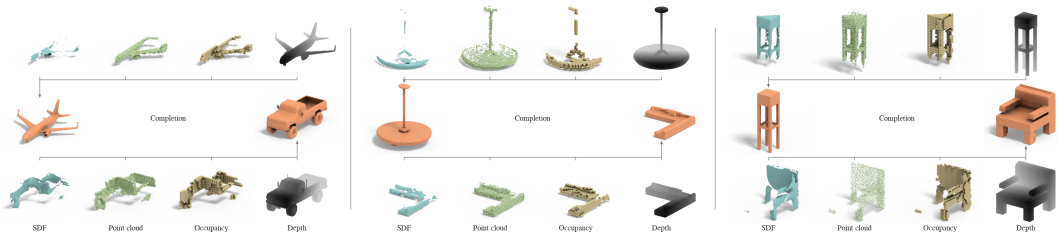


Figure 1: Multimodal Shape Completion. Each block shows the shape completion from different partial modalities of the same object.

ABSTRACT

Existing shape completion methods accept a single input modality and, when built on dense voxel grids, deliver watertight surfaces only at low resolution. Latent vector sets decouple the latent from the grid, but their latent tokens come from input-dependent queries, lack of consistent correspondence between partial and complete shapes, and surface extraction still needs a dense field query. We present UniComplete, a framework for multimodal shape completion built on an anchored latent set: each token is bound to a fixed canonical 3D anchor, so it keeps the same identity and spatial reference across shapes, modalities, and completeness levels. An SDF VAE defines the shared latent space, and modality-specific encoders are distilled into it, so signed and unsigned distance fields, occupancy grids, point clouds, and depth maps share one decoder and one completion model. Completion is a posterior-mean rectified flow: a partial encoder regresses the posterior-mean estimate of the complete latent, and a rectified flow started from that estimate yields multiple plausible completions in a few steps. Sparse surface encoding and a coarse-to-fine Surface Gate confine computation to the surface at the input and the output, so cost scales with the surface rather than the volume. We further introduce SfMComplete, a benchmark pairing partial shapes reconstructed by structure-from-motion from rendered images with exact CAD ground truth across modalities and resolutions. UniComplete is competitive on 3D-EPN and PatchComplete, improves over compared baselines on SfMComplete at higher resolutions, and completes all five input modalities at SDF resolutions up to 512^3.

1 INTRODUCTION

Scanning pipelines never hand over a complete shape: a depth camera yields depth maps with occlusions, structure-from-motion (SfM) on a set of images returns a sparse cloud with floaters, and LiDAR returns one sweep of sparse points. Downstream tasks such as contact- and physics-based grasp planning Varley et al. (2017), collision and simulation in real-to-sim pipelines Dai et al. (2024), and rigging, animation, and editing of 3D assets Baran & Popović (2007) require watertight shapes. Shape completion, recovering the full object shape from its partial scan, is the step in between. It is inherently ill-posed, since the regions a scan did not observe are consistent with many plausible complete shapes; discriminative methods regress a single completion Dai et al. (2017b); Rao et al. (2022); Yu et al. (2021), while generative methods model the distribution of complete shapes conditioned on the partial input, from GAN and autoregressive priors Wu et al. (2020); Mittal et al. (2022) to diffusion and flow matching Chu et al. (2023); Cui et al. (2024); Kong et al. (2026).

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.

Existing completion methods, discriminative or generative, share two limits set by their shape representation: the modality they support and the resolution at which they can deliver a watertight surface. First, prior methods take the partial observation in a single modality. Where further inputs are accepted, they are extra scans in the same encoding Chu et al. (2023), prompts Cheng et al. (2023); Aiello et al. (2022), or coarse hints of the complete shape rather than partial observations Chibane et al. (2020); Zhang et al. (2024). To our knowledge, no method completes partials given as signed or unsigned distance fields (UDF or SDF), occupancy grids, point clouds, or depth maps within one framework. Second, the representation decides the resolution at which a watertight surface can be delivered. Dense voxel grids Chu et al. (2023); Kong et al. (2026) grow cubically in memory and compute, which caps input and output at 32^3 to 128^3 and spends most of that cost on empty space, while point-cloud completion Yuan et al. (2018); Yu et al. (2021) avoids the grid but returns sparse unstructured points, from which a watertight surface is recovered at a loss of geometric fidelity.

Latent vector sets Zhang et al. (2023) lift the resolution limit. 3DShape2VecSet and its extensions Li et al. (2025a); Cho et al. (2025); Chen et al. (2025) encode a shape as a fixed-size set of latent vectors that decode a continuous neural field, so the latent no longer grows with the grid, although extracting a surface at high resolution still costs a dense pass over the output. However, the vectors come from unordered, input-dependent queries, so the vector at a given index has no consistent meaning across shapes, and the latents of a partial shape and its completion, or of two encodings of the same object, cannot be aligned token by token.

To address both limits coherently, we present UniComplete, a framework that encodes shapes of varying completeness and input modalities into a shared latent space through a multi-encoder VAE, and completes the partial shapes by a partial-to-complete transport in that space.The core is an anchored latent: a set of latent vectors, each bound to a fixed canonical 3D position that is shared by every shape, modality, and completeness level. Each token therefore keeps a fixed identity and spatial reference across inputs, so latents can be compared and aligned token by token.

This correspondence buys two things. A new input encoding becomes an encoder-only addition: we train an anchored SDF VAE once, freeze its latent space and decoder, and distil each further encoder into the same latent. And completion becomes a transport that separates accuracy from plausibility: a partial encoder regresses the posterior-mean estimate of the complete latent Freirich et al. (2021); Ohayon et al. (2025), and a rectified flow Liu et al. (2023b) launched from that estimate carries it to the distribution of complete latents in a few Euler steps, one transport for every partial modality.

The anchored latent has a fixed size whatever the grid, but encoding and decoding at high resolution must still be affordable, and we make them so by processing only the surface. At the input, each encoder reads only surface features, the near-surface band of a voxel grid through sparse convolution Yan et al. (2018), sampled points, or DINO features of a depth map Oquab et al. (2024). At the output, a coarse-to-fine Surface Gate confines the dense SDF queries of the triplane decoder Cho et al. (2025) to surface-bearing cells. Both costs therefore scale with the surface rather than the volume, which takes our method to 512^3 on a single GPU.

Existing benchmarks cannot measure either limit. PCN Yuan et al. (2018) and ShapeNet-55 Yu et al. (2021) are point-cloud only, and 3D-EPN Dai et al. (2017b) and PatchComplete Rao et al. (2022) provide 32^3 TSDF partials with complete shapes up to 128^3; their partials come from synthetic depth views with exact poses and clean depth, so they lack the noise of an actual reconstruction pipeline, and the real ScanNet split of PatchComplete has only approximate ground truth. We therefore introduce SfMComplete. Its partial shapes are reconstructed by SfM from images rendered from ShapeNet objects, so they carry the holes, uneven density, and floaters while the ground truth remains the exact CAD model; every partial is provided in all five modalities and at every resolution, so completion can be evaluated across modalities and resolutions on the same objects.

UniComplete ranks first on SfMComplete and is competitive with specialised methods on 3D-EPN and PatchComplete, and completes all five modalities through one transport and one decoder at resolutions up to 512^3 on a single GPU.

Our contributions are as follows.

- **A unified latent space for shape reconstruction and completion.** The vector-set latent is bound to fixed canonical anchors, so latents of a partial shape and its completion, and of different modalities, correspond by construction rather than by training.

- **A posterior-mean-then-transport completion pipeline.** We decouple accuracy from plausibility: a regression-trained partial encoder supplies the distortion-optimal estimate of the complete latent, and a condition-centred rectified flow started from that estimate turns it into multiple plausible completions in a few steps (Sec. 3.3).
- **Multi-modal shape reconstruction and completion with encoder-only extension.** SDF, UDF, occupancy grids, point clouds, and depth maps share one transport and one decoder, and a new input encoding requires only a new encoder trained into the frozen latent by feature-level distillation (Sec. 3.2).
- **A surface-only VAE for high-resolution completion.** Sparse surface encoders, a coarse-to-fine Surface Gate, and surface-concentrated supervision confine computation to the surface at the input, the output, and the loss, so one fixed-size latent serves resolutions that dense-voxel completion cannot reach on a single GPU (Sec. 3.1).
- **A benchmark for pipeline-realistic completion.** SfMComplete pairs partial shapes from an SfM reconstruction of rendered images with exact CAD ground truth, in all five partial modalities and at 64^3 to 512^3, so that the modality and resolution axes are measurable.

2 Related Work

3D shape completion. Early methods regress a single completion from a partial grid or point cloud (Dai et al., 2017b; Chibane et al., 2020; Peng et al., 2020; Rao et al., 2022; Yuan et al., 2018; Yu et al., 2021; 2023; Chen et al., 2023; Zhang et al., 2026). Since a partial scan admits many completions, later work learns a generative prior, from GANs and autoregressive models (Wu et al., 2020; Mittal et al., 2022; Yan et al., 2022; Zakeri et al., 2025) to diffusion in voxel or latent space (Cheng et al., 2023; Chu et al., 2023; Schaefer et al., 2024; Cui et al., 2024), all starting from noise with the partial injected as a condition or by inpainting; BridgeShape (Kong et al., 2026) instead bridges from a partial latent, trained into a frozen VQ-VAE, to the complete one. Grid-based methods pay for every voxel; sparse convolution (Yan et al., 2018; Ren et al., 2024; Li et al., 2025b) at the input and coarse-to-fine extraction (Mescheder et al., 2019; Tatarchenko et al., 2017; Wang et al., 2018; Takikawa et al., 2021; Müller et al., 2022) at the output avoid that cost. UniComplete keeps BridgeShape's latent-to-latent view in a fixed-size set latent rather than a VQ grid, and applies both remedies to completion.

Set latents and token identity. 3DShape2VecSet (Zhang et al., 2023) encodes a shape as a set of latent vectors decoding a continuous field, the representation behind most large 3D generators (Zhang et al., 2024; Li et al., 2025a; Zhao et al., 2025; Chen et al., 2025; Cho et al., 2025). What a token index means depends on the query that produced it: learnable queries (Zhang et al., 2023) fix identity but carry no position, farthest-point queries (Zhang et al., 2024; Li et al., 2025a; Chen et al., 2025; Cho et al., 2025) and irregular latent grids (Zhang et al., 2022a) fix positions but resample them per input, and sparse-voxel latents (Ren et al., 2024; Xiang et al., 2024; Li et al., 2025b) yield an input-dependent token set. None gives token i both a fixed identity and a fixed position across a partial shape, its completion, and other encodings of the same object. Our anchored latent fixes both identity and position by binding every token to a canonical anchor shared by all inputs; unlike the per-input learned nodes of AnchorFormer (Chen et al., 2023), the anchors never move.

Multiple input modalities in one latent. Aligning encoders of different modalities to a shared code is established for retrieval and generation, as a global embedding aligned with images and text (Xue et al., 2023; Zhao et al., 2023) or a code reachable from voxels, points, or views (Muralikrishnan et al., 2019), but not for completion. Completion methods that accept several inputs treat the extras as auxiliary signals: RGB images beside a partial point cloud (Zhang et al., 2021; Aiello et al., 2022), text or images as prompts or as condition branches on a noise-source generator (Cheng et al., 2023; Cui et al., 2024; Schaefer et al., 2024; Zhang et al., 2024; Hui et al., 2024), or points discretised onto a voxel grid with one network per input setting (Chibane et al., 2020). UniComplete instead distils each modality encoder into a frozen anchored latent, so voxel grids, point clouds, and depth maps enter the same transport and decoder as interchangeable partial inputs.

Paired transports and posterior-mean estimates. Flow matching and rectified flow (Lipman et al., 2023; Liu et al., 2023b) regress a velocity along straight interpolants. Stochastic interpolants with data-dependent couplings (Albergo et al., 2023; 2024), InDI (Delbracio & Milanfar, 2023), I^2SB (Liu et al., 2023a), and diffusion bridges (De Bortoli et al., 2021; Shi et al., 2023; Zhou et al.,

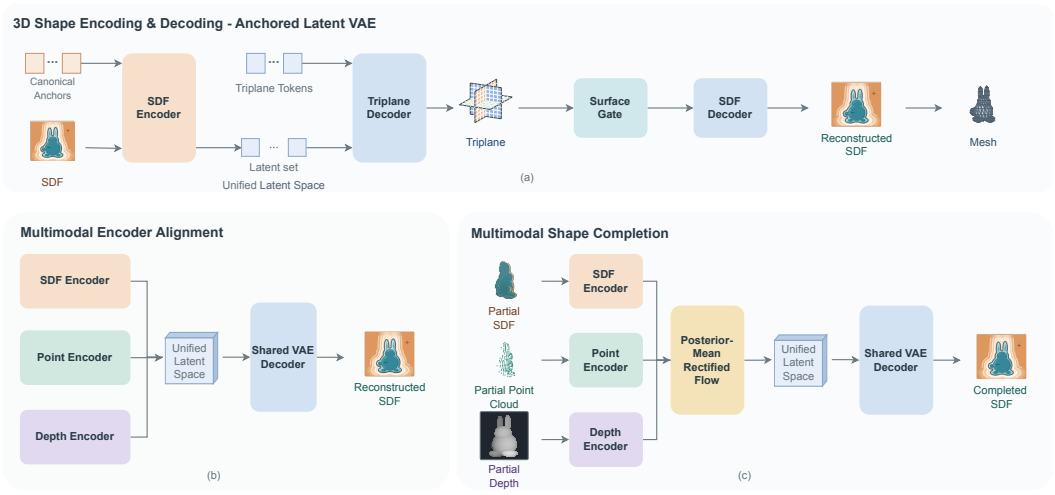


Figure 2: **UniComplete.** (a) Anchored SDF VAE (Sec. 3.1): the VAE takes SDF and canonical anchors as input and reconstructs the SDF. (b) Multimodal latent alignment (Sec. 3.2): each modality's encoder is aligned to the latent space defined by the anchored SDF VAE. (c) Posterior-mean rectified flow (Sec. 3.3): a partial encoder supplies the posterior-mean estimate, and a rectified flow with source noise carries it to plausible complete latents for SDF decoding.

2024; Chadebec et al., 2025) replace the Gaussian source with a degraded observation, the construction our transport uses. Since the posterior mean is the distortion-optimal estimate and sample quality must be traded against it (Freirich et al., 2021), PMRF (Ohayon et al., 2025) first regresses the posterior mean and then flows from it to the data manifold. UniComplete applies the same two stages in the anchored latent, where partial and complete shapes are paired by construction.

3 Method

UniComplete rests on one core design: the tokens of a set latent are bound to fixed canonical 3D anchors, so that latents of any shape, modality, or completeness level correspond (Fig. 2). Sec. 3.1 defines this anchored latent and its VAE; Sec. 3.2 maps every modality onto the same anchors; Sec. 3.3 completes a shape by transporting the partial estimate to the distribution of complete latents; Sec. 3.4 makes high resolution affordable by surface-focusing computation at the input, output, and loss. The components are trained in order, each on the frozen output of the previous (Appx. B).

3.1 Anchored Latent Set

Canonical anchors. A set latent (Zhang et al., 2023) represents a shape as M vectors $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^{M}$, $\mathbf{z}_i \in \mathbb{R}^D$, obtained by letting M queries cross-attend to features of the input surface; the query decides what index i means. We fix *canonical anchors* $\{\mathbf{a}_i\}_{i=1}^{M} \subset \mathbb{R}^3$, computed once by k-means over surface points pooled across the training set and shared by every shape, modality, and completeness level; token i is initialised from the positional embedding of $\mathbf{a}_i$ and refined by cross-attention over all input surface features. The latent is therefore global: the anchor fixes the token's identity and spatial reference, not the region it locates, so latents of different inputs can be compared, aligned, and transported, and a partial observation still fills every anchor from the observed surface rather than leaving holes. The price is the per-shape adaptivity of farthest-point queries, which Sec. 4 measures on reconstruction. We use M=40 and D=32 regardless of the input or output grid.

Anchored Latent VAE. The *anchored encoder* takes surface tokens, each a position in the canonical frame with a feature vector (Sec. 3.2), and follows COD-VAE's progressive design (Cho et al., 2025): the anchors read from patches of tokens and then from the tokens, and a KL block compresses the working width C=512 to D and regularises $\mathcal{Z}$ toward a unit Gaussian. The decoder expands $\mathcal{Z}$ to width C, populates a triplane with transformer blocks, and reads a signed distance at any query point through a shallow MLP on sampled triplane features; marching cubes (Lorensen & Cline, 1987) then

yields a watertight mesh at any resolution V. We predict signed distance rather than occupancy for its sub-voxel zero level set and gradient information (Park et al., 2019a).

3.2 Multimodal Latent Alignment for Shape Reconstruction

The anchored latent is defined by the SDF VAE, whereas a scanning pipeline can deliver distance grids, occupancy grids, point clouds, or depth maps. We therefore build a multi-encoder VAE: the encoder of each modality is aligned to the SDF teacher's latent of the same complete shape, an alignment well definedd by the shared anchors, so one decoder can be shared by any input.

Multimodal surface Encoder. Each modality has its own surface encoder. *Voxel grids:* SDF, UDF, and occupancy grids share the sparse surface encoder of Sec. 3.4: the position is the centre of the surface voxel, the feature its sparse-convolutional output. *Point clouds:* The position is the point, the feature its Fourier embedding, as in COD-VAE. *Depth maps:* DINOv2 (Oquab et al., 2024) encodes the foreground depth into patch tokens: the position is the patch centre unprojected with the known intrinsics, the feature the patch token.

Multi-encoder VAE. The SDF VAE, trained first on complete shapes, is the teacher; each further modality m gets its own copy of the anchored encoder, E^m, and only its surface encoder and E^m are trained, so that for the same complete shape E^m lands on the teacher's latent. With $\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i \in \mathbb{R}^D$ the student's posterior mean and standard deviation at anchor i, and starred quantities the frozen target,

$$\begin{aligned}\mathcal{L}_{\text{align}} = \lambda_1 \frac{1}{MD} \sum_{i=1}^{M} \big\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_i^\star\big\|_1 + \lambda_{\text{rel}} \frac{1}{M^2 \bar{d}^2} \sum_{i,j=1}^{M} \Big(\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2 - \|\boldsymbol{\mu}_i^\star - \boldsymbol{\mu}_j^\star\|_2\Big)^2 \\ + \lambda_{\text{kl}} \sum_{i=1}^{M} \mathrm{KL}\Big(\mathcal{N}\big(\boldsymbol{\mu}_i, \operatorname{diag}\boldsymbol{\sigma}_i^2\big) \,\Big\|\, \mathcal{N}\big(\boldsymbol{\mu}_i^\star, \operatorname{diag}\boldsymbol{\sigma}_i^{\star 2}\big)\Big) + \lambda_{\text{sdf}}\, \mathcal{L}_{\text{SDF}}\big(\mathrm{Dec}(\boldsymbol{\mu})\big),\end{aligned} \tag{1}$$

where $\bar{d}^2 = \frac{1}{M^2}\sum_{i,j}\|\boldsymbol{\mu}_i^\star - \boldsymbol{\mu}_j^\star\|_2^2$ makes the relational term scale-free. The ℓ_1 term pulls each token to the teacher's, the relational term preserves pairwise token distances within a shape (Park et al., 2019b), the KL term matches the posterior rather than only its mean, and the SDF term ties the alignment to geometry, which latent terms cannot since the decoder is nonlinear and many-to-one.

3.3 Completion as a posterior-mean rectified flow

A partial scan is a strong but imperfect condition: it captures the shape wherever the sensor looked, noisy and incomplete. Generative methods that start from Gaussian noise and use the scan only as a condition (Chu et al., 2023; Cui et al., 2024; Schaefer et al., 2024) ignore this structure and pay twice: the transport has to cross the whole latent space along a curved velocity field, which costs steps (Liu et al., 2023b), and everything the scan already fixes has to be rediscovered through attention. Starting at the condition shortens the path, but by how much depends on where exactly one starts: BridgeShape (Kong et al., 2026) starts from the raw partial latent plus noise, a reconstruction code of the scan with its holes and errors, so everything the scan missed or got wrong is still left for the transport to infer while it samples. We start closer. The conditional mean $\mathbb{E}[\mathbf{z}_{\text{c}} \mid \text{partial}]$ of the complete latent $\mathbf{z}_{\text{c}}$ is the most accurate completion any method can return if it returns one shape per scan, and it is blurry wherever the scan leaves the shape uncertain, so anything sharper pays in accuracy (Freirich et al., 2021). Following PMRF (Ohayon et al., 2025), we first close the determinable part of the partial-to-complete gap with a regression on partial–complete pairs and let a rectified flow with source noise resolve only what the partial leaves undetermined; as the source noise vanishes, the result is at least as accurate as sampling from the posterior (Ohayon et al., 2025).

Posterior-mean estimate. The posterior-mean estimate comes from a partial encoder. For each modality, the partial encoder is a copy of its complete encoder E^m, fine-tuned on partial–complete pairs with the alignment loss Eq. (1) and the frozen SDF latent of the complete shape. The regression learns the point estimate with the least expected error, which is the conditional mean: $\mathbf{z}_{\text{p}} \approx \mathbb{E}[\mathbf{z}_{\text{c}} \mid \text{partial}]$, where $\mathbf{z}_{\text{p}}$ is the posterior-mean latent.

Posterior-mean rectified flow. Given a pair $(\mathbf{z}_{\text{p}}, \mathbf{z}_{\text{c}})$, we define the flow state

$$\mathbf{x}_t = (1-t)\,(\mathbf{z}_{\text{p}} + \alpha\,\boldsymbol{\epsilon}) + t\,\mathbf{z}_{\text{c}}, \qquad t \in [0,1],\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),\ \alpha \geq 0, \tag{2}$$

which starts at $t{=}0$ from $\mathbf{z}_\mathrm{p}$ perturbed by Gaussian noise of scale α and reaches $\mathbf{z}_\mathrm{c}$ at $t{=}1$. Because the noise enters only at the source, every training path is a straight line with constant velocity $\mathbf{z}_\mathrm{c} - \mathbf{z}_\mathrm{p} - \alpha\boldsymbol{\epsilon}$. The noise also decides what the flow can return: without it the source is a deterministic function of the scan and the flow returns one completion per scan; with it the same $\mathbf{z}_\mathrm{p}$ is carried to different complete latents, at an accuracy cost that grows with α, so the noise scale is a single knob between fidelity and diversity. The velocity network v_θ is a variant of TripoSG DiT (Li et al., 2025a) whose tokens carry a positional encoding of their anchors, trained with

$$\mathcal{L}_\mathrm{FM} = \mathbb{E}_{(\mathbf{z}_\mathrm{p},\mathbf{z}_\mathrm{c}),\, t\sim\mathcal{U}[0,1],\,\boldsymbol{\epsilon}} \left\| v_\theta(\mathbf{x}_t, t, \mathbf{z}_\mathrm{p}) - (\mathbf{z}_\mathrm{c} - \mathbf{z}_\mathrm{p} - \alpha\boldsymbol{\epsilon}) \right\|_2^2, \tag{3}$$

with α drawn uniformly from $[0, 1]$ per sample during training, so the flow is trained at every noise scale. $\mathbf{z}_\mathrm{p}$ can optional enter as cross-attention conditioning as in DepthFM (Gui et al., 2025) but it helps marginally in our setting. At inference we draw $\boldsymbol{\epsilon}$, start from $\mathbf{x}_0 = \mathbf{z}_\mathrm{p} + \alpha\boldsymbol{\epsilon}$, integrate v_θ from $t{=}0$ to 1 in a few Euler steps, and decode; α therefore acts as a sampling temperature. The anchored latent makes a posterior-mean source well posed for shapes and its unified latent space makes this partial-to-complete transport shared among different modalities.

3.4 Surface-only computation

Dense-grid completers (Dai et al., 2017b; Chu et al., 2023) process every voxel of a V^3 grid, so their cost grows cubically with V, and even set-latent VAEs such as COD-VAE still evaluate the decoded field at every voxel. Since a shape's geometry lives around its surface, we restrict computation to the surface at the input (surface encoder), the output (Surface Gate), and the loss (surface-concentrated supervision), so cost scales with the surface rather than the volume.

Surface encoder for voxel grids (input). A shape in voxel grid is determined by its surface, so we design the surface encoder reads only a 3-voxel band centred on the zero-crossing voxels. A stack of sparse convolution–downsampling blocks (Yan et al., 2018), as many as the input resolution needs to reach a fixed coarse resolution, turns this band into surface features, each a convolutional feature plus a positional embedding of its voxel centre. Surfaces differ in size, so we pad the features to the batch maximum and let the anchors read them through masked cross-attention.

Surface Gate (output). Meshing requires a full grid, over 10^8 at $V{=}512$, but most voxels are empty. A small classifier head on the frozen triplane features predicts *which* cells of a coarse V_c^3 grid ($V_c \in \{32, 64\}$) contain surface, a far easier task than SDF querying; a tuned threshold and a one-cell dilation guard against misses. At inference only the kept cells are subdivided to resolution V and queried, while all other voxels receive a constant interior or exterior value; Sec. 4 confirms that this recovers every sign crossing of the dense evaluation while removing over 80% of the queries.

Surface-concentrated supervision (loss). COD-VAE spends a fixed budget of $|\mathcal{Q}|$ query points per iteration; we stratify it over the surface, the near-surface band (balanced inside and outside), and the outer region, so most supervision lands where the field is hardest to learn. The loss pairs a per-region ℓ_1 regression to the truncated SDF with a sign-aware term, binary cross-entropy in the band and a hinge outside, that catches the sign flips ℓ_1 barely sees, plus normal and eikonal regularisers at surface points (Gropp et al., 2020) with per-term definitions and weights in Appx. C.:

$$\mathcal{L}_\mathrm{SDF} = \lambda_\mathrm{r}\mathcal{L}_\mathrm{recon} + \lambda_\mathrm{b}\mathcal{L}_\mathrm{BCE} + \lambda_\mathrm{h}\mathcal{L}_\mathrm{hinge} + \lambda_\mathrm{n}\mathcal{L}_\mathrm{normal} + \lambda_\mathrm{e}\mathcal{L}_\mathrm{eik}, \tag{4}$$

4 Experiments

4.1 Experimental Setup

We evaluate on 3 benchmarks: 3D-EPN (Dai et al., 2017b) for known classes, PatchComplete (Rao et al., 2022) for unseen classes on synthetic and real scans, and our proposed SfMComplete for realistic partial observations across modalities and resolutions. Chamfer Distance(**CD**) is scaled by 10^2, IoU and F1 are percentages, and per-column rankings are marked first, second, and third.

3D-EPNpairs 6 virtually scanned 32^3 TSDF partials of each ShapeNet (Chang et al., 2015) shape in 8 categories with a complete TUDF at 32^3 for benchmark, scored by the **mean** ℓ_1 **error** of the predicted UDF. **PatchComplete** tests unseen categories on a synthetic ShapeNet split and a real split

cropped from ScanNet (Dai et al., 2017a) with Scan2CAD (Avetisyan et al., 2019) ground truth, all 32^3 TSDFs, scored by ℓ_1 **CD** and volumetric **IoU**.

SfMComplete. 3D-EPN and PatchComplete synthetic benchmarks virtually scan a complete CAD model, so their partials lack the artefacts of a real pipeline, while in-the-wild captures, PatchComplete real and ShapeR (Siddiqui et al., 2026) pair realistic partials with approximate ground truth (GT). SfMComplete has both: its partials are reconstructed by feature-based SfM from image trajectories rendered around ShapeNet objects of the 8 3D-EPN classes, about 35K objects and 140K partials, so they carry the noise of a real pipeline while the GT stays the exact CAD model. Every partial comes as a SDF, UDF, occupancy grid, point cloud, and depth map. Voxel grid partial and GT are at 64^3 to 512^3, so reconstruction and completion can be evaluated across modalities and resolutions (Tab. 9). Construction pipeline and comparison to other datasets are in Appx. D.

We report **CD**, **IoU**, and **F1@0.01** of the reconstructed and completed shape against the GT. For generative completion methods, every stochastic method draws $K{=}5$ completions per partial and each metric is reported twice: **best of** K measures the quality a method can reach, and **mean of** K measures the average quality it delivers on a single draw. **TMD** (Wu et al., 2020), the average pairwise CD among the K completions, measures the diversity.

Implementation. The anchored VAE uses $M{=}40$, $D{=}32$, and width $C{=}512$; the flow is an 11-layer TripoSG-variant DiT; all stages are trained with Adam at 10^{-4} in mixed precision on $8{\times}$RTX A6000. Benchmark statistics and further training details are in Appx. E. For the evaluation, we run 2 Euler steps with $\alpha = 1$ for partial-to-latent generation. The details of the benchmarked methods on our dataset are in Appx. E

4.2 Experiment results on 3D-EPN and PatchComplete

Table 1: Shape completion on 3D-EPN (*seen* categories): mean ℓ_1 error averaged over categories.

Metric	3D-EPN	ConvONet	SDF-StyleGAN	cGCA	AutoSDF	ShapeFormer	PatchComplete	DiffComplete	BridgeShape	Ours
$\ell_1\downarrow$	0.374	0.220	0.278	0.185	0.217	0.141	0.088	0.053	0.039	0.043

Table 2: Shape completion on PatchComplete on *synthetic* and *real-world* splits, IoU and CD averaged over the categories.

Split	Metric	3D-EPN	Few-Shot	IF-Nets	AutoSDF	ConvONet	PatchComplete	DiffComplete	BridgeShape	Ours
Synthetic	CD↓	5.58	9.58	5.29	5.86	5.26	4.27	4.10	4.06	4.10
	IoU↑	59.4	40.3	58.1	45.2	60.1	65.4	67.5	70.1	67.7
Real-world	CD↓	9.09	9.02	8.26	8.90	8.34	7.52	7.18	6.99	7.01
	IoU↑	44.0	38.6	42.6	38.9	42.9	49.8	51.3	53.5	52.0

We compare against 3D-EPN, ConvONet, SDF-StyleGAN (Zheng et al., 2022), cGCA (Zhang et al., 2022b), AutoSDF, ShapeFormer, Few-Shot (Wallace & Hariharan, 2019), IF-Nets, PatchComplete, DiffComplete, and BridgeShape; Tab. 1 and Tab. 2 report category means, and Appx. F.1 shows the per-category results. On both 32^3 benchmarks UniComplete is second only to BridgeShape and matches or beats all others. The three methods differ on two axes: DiffComplete runs a 3D CNN on the 32^3 grid and BridgeShape on a 16^3 VQ latent of it, while we use a transformer on 40 anchored vectors; and BridgeShape and we transport from the partial latent, whereas DiffComplete denoises from Gaussian noise with the partial as a condition. BridgeShape holds both advantages and is best at 32^3; DiffComplete holds only the grid and we have the start. The partial start counts for more because it changes what the generator must do while the grid only changes how faithfully it does it. What separates us from BridgeShape is the dense 16^3 VQ latent. Our 40 tokens compress the same grid ten times more and cannot match a near-lossless local code there. That edge lasts only while the grid is compact and higher resolution turns in our favour as shown in (Tab. 3). The gap also shrinks from 10% in ℓ_1 on the seen categories of 3D-EPN to 1% and 0.3% in CD on the unseen categories of PatchComplete: the global anchored latent generalizes better than a grid code.

4.3 Experiment results on SfMComplete

SfMComplete tests what the standard benchmarks cannot: multimodal shape reconstruction and completion at multiple resolutions.

Completion against matched baselines. Tab. 3 compares UniComplete with baselines at the modality and resolution it supports. UniComplete leads every completion column, and the margin widens from best of K to mean of K: every draw departs from the same posterior-mean estimate, so

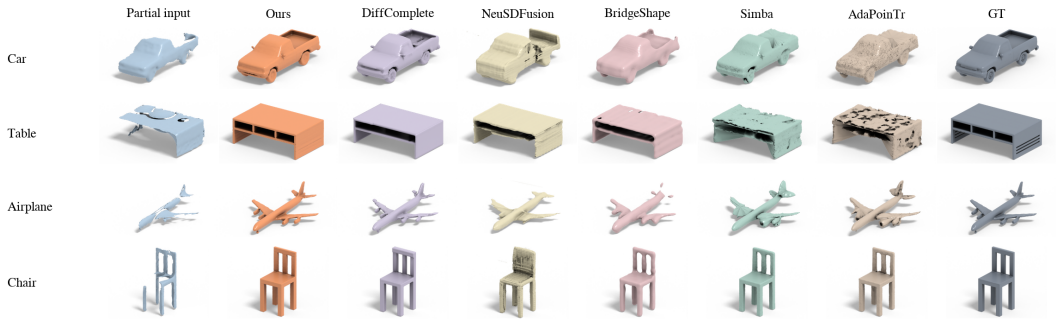


Figure 3: Qualitative results of shape completion on SfMComplete. The partial is an SfM reconstruction; we show its SDF, and each baseline runs on the same object in the modality it supports.

the draws stay concentrated, and since the best draw leads as well they are concentrated on the right shape rather than collapsed onto a wrong one. The price is diversity, where our TMD is the lowest. The gain is not inherited from the autoencoder: BridgeShape reconstructs the complete shape better than we do at 64^3 yet completes it worse, and at 128^3 its VQ grid drops sharply in both reconstruction and completion where our anchored latent holds steady; DiffComplete and BridgeShape run out of memory at 128^3 and 256^3. On point clouds, with only the encoder swapped and the same transport and decoder, UniComplete-PC leads NeuSDFusion by a wide margin, and its average draw beats the best output of AdaPoinTr and Simba. Fig. 3 shows the qualitative results. Pipeline-realistic partials are harder than the synthetic scans of 3D-EPN for every method, as the benchmark intends.

Resolution and modality. Tab. 4 varies the grid from 64^3 to 512^3 with SDF input to test scalability (left) and the input modality at 128^3 to test multimodal completion (right). Across resolutions the latent keeps its 40 tokens while the grid grows, and completion degrades gracefully: it peaks at 128^3 and gives up only a couple of IoU points by 512^3, less than half the drop in reconstruction, so completion is less sensitive to resolution than the encoder itself. Across modalities only the encoder changes (Sec. 3.2). The alignment columns score the partial after each encoder maps it into the shared latent, and the completion columns the transport run from there: the four encoders that see the same SfM partial in different modalities align within little more than a point of one another. The transport lifts each of them above its alignment score, and all four complete within a point of the SDF teacher in the same order, so one frozen transport and decoder serve every modality without modality-specific training. The depth row is not comparable, since its partial is a clean rendered depth view rather than the noisy SfM partial, which is why it scores highest.

Table 3: Shape completion on SfMComplete, matched per modality and resolution.

			Best of K			Mean of K				
Method	Input	In/Out res.	IoU↑	CD↓	F1@0.01↑	IoU↑	CD↓	F1@0.01↑	TMD↑ ($\times 10^{-3}$)	Rec. IoU↑
DiffComplete	SDF	$64^3/64^3$	52.45	5.51	81.33	40.11	6.74	74.75	32.65	–
BridgeShape	SDF	$64^3/64^3$	64.47	4.63	86.40	60.81	4.92	84.12	5.15	92.08
UniComplete-SDF (Ours)	SDF	$64^3/64^3$	65.58	4.52	86.89	64.69	4.76	85.69	2.65	86.43
DiffComplete	SDF	$128^3/128^3$				OOM				
BridgeShape	SDF	$128^3/128^3$	59.53	4.98	83.51	56.48	5.33	80.98	5.82	72.60
UniComplete-SDF (Ours)	SDF	$128^3/128^3$	64.09	4.41	87.72	62.92	4.54	86.83	3.48	88.43
BridgeShape	SDF	$256^3/256^3$				OOM				
NeuSDFusion	point cloud	$-/256^3$	45.71	9.13	62.75	43.07	10.14	59.22	15.61	66.87
UniComplete-PC (Ours)	point cloud	$-/256^3$	63.39	4.74	86.12	62.14	4.82	86.12	2.93	85.21
AdaPoinTr†	point cloud	–/–	54.07	4.93	83.32	54.07	4.93	83.32	0	–
Simba	point cloud	–/–	56.72	4.72	84.76	53.68	5.04	83.00	5.97	–

Table 4: Resolution scaling (SDF input, left) and input modality at 128^3 (right) on SfMComplete; the decoder and transport are the SDF-trained ones in every row.

	VAE recon.			*Completion (Best-of-K)*				*Partial Alignment*			*Completion (Best-of-K)*		
Resolution	IoU↑	CD↓	F1@0.01↑	IoU↑	CD↓	F1@0.01↑	*Modality*	IoU↑	CD↓	F1@0.01↑	IoU↑	CD↓	F1@0.01↑
64^3	86.43	3.24	96.33	65.58	4.52	86.89	SDF	63.26	4.53	86.96	64.09	4.41	87.72
128^3	88.43	2.84	97.64	64.09	4.41	87.72	UDF	64.14	4.44	87.56	64.79	4.35	88.22
256^3	84.69	3.36	95.86	63.55	4.63	86.10	Occ	62.81	4.56	86.74	63.40	4.47	87.35
512^3	82.52	3.42	94.47	61.83	4.94	85.12	PC	63.77	4.53	87.00	64.72	4.37	88.08
							Depth	74.01	3.68	91.99	73.95	3.66	92.21

4.4 Ablations

Surface Gate. Tab. 6 shows that the Surface Gate keeps the output mesh intact while saving over 80% of the SDF queries from the triplane. The gate runs on a 32^3 or 64^3 grid for SDFs at 128^3 and 256^3; we report *recall*, the percentage of surface voxels that the gate can detect, and *saving*, the portion of queries saved. Without dilation the gate already recalls over 99% while saving over 90%; the one-cell dilation we use recalls every crossing and still saves over 80%.

Anchor scheme. Tab. 7 shows the reconstruction and completion results of different latent structure choices. The FPS tokens have the best reconstruction performance due to its adaptivity to the inputs but it can only be a cross-attention condition, which delivers worst completion. Learnable queries can be aligned but it lacks spatial reference, and thus underperforms our canonical anchors.

Partial-to-complete transport. Tab. 8 swaps only the transport on the same anchored latents: a noise-source flow with z_{p} as cross-attention condition, a Brownian bridge from z_{p} (Liu et al., 2023a; Kong et al., 2026), and our posterior-mean rectified flow (Sec. 3.3). Starting the flow from z_{p} benefits. Our source-only noise keeps the velocity target constant and the path straight; the bridge's mid-path noise makes the target high-variance at the endpoints and the path curved, so our choice has a small win.

Table 5: Few-step Euler convergence of the posterior-mean RF (128^3 SDF).

Metric	Euler steps (NFE) 1	2	4	8	16
IoU↑	63.86	64.19	64.12	63.96	63.83
CD↓	4.427	4.398	4.399	4.410	4.420
F1@0.01↑	87.63	87.85	87.81	87.72	87.64

Table 6: Effective ablation of Surface Gate on SfMComplete

SDF res. (R)	Gate res. (V_c)	No dilation Recall%↑	No dilation Saving%↑	1-cell dilation (default) Recall%↑	1-cell dilation (default) Saving%↑
128^3	32	99.80	90.56	100.00	81.61
256^3	32	99.66	91.68	100.00	82.70
256^3	64	99.31	92.74	100.00	87.79

Table 7: Anchor-scheme ablation on SfMComplete (128^3 SDF).

Anchor scheme	Condition	Recon IoU↑	Compl. IoU↑
FPS point queries	x-attn	92.0	52.28
Learnable queries	start point	87.48	56.64
Canonical anchors (Ours)	start point	88.43	64.09

Table 8: Partial-to-complete transport ablation (64^3 SDF).

Transport design	IoU↑	CD↓	F1@0.01↑
X-attn cond	60.36	5.12	82.37
Brownian bridge	64.54	4.71	85.19
Posterior-mean RF (Ours)	65.58	4.52	86.89

Few-step Euler convergence. Tab. 5 shows the complete results over the Euler steps from 1 to 16. The RF converges at 2 steps which turns to be our default.

5 Conclusion

We presented UniComplete, a multimodal shape-completion framework built on a latent space whose tokens are bound to fixed canonical anchors, so that the latents of different modalities and completeness levels correspond token by token. That correspondence enables the rest of the method: a new input modality enters by distilling its encoder into the latent of the frozen SDF VAE and completion becomes a posterior-mean rectified flow that lands in a single Euler step. The surface-only encoding and the Surface Gate, make the method easily scale to 512^3 on a single moderate GPU. Although UniComplete is the second in two 32^3 benchmarks, nn SfMComplete, the pipeline-realistic, multimodal, and multi-resolution benchmark we introduce, UniComplete leads every matched baseline at its own modality and resolution. It also demonstrate the ablity of our method to scale to high resolution grids and handles multimodal inputs within one framework.

Limitations and future work. The anchors are fit by k-means to object-level surfaces in a canonical frame, so the latent assumes a single, roughly aligned object and scene-level and open-vocabulary completion are untested. The fixed-size latent that makes 512^3 affordable also sets a ceiling: reconstruction and completion peak at 128^3 and decline slowly at finer grids, and at the 32^3 grids of 3D-EPN a near-lossless grid code still wins on per-voxel error. Starting from the posterior mean concentrates the draws, which buys accuracy at the cost of diversity, and the noise scale α is the only knob between them. Two directions follow: anchors fit per region rather than per object, to carry the latent to partial scenes, and a larger SfMComplete with more object classes.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code development, research idea analysis, literature review, and manuscript polishing.

We did not use generative AI tools to generate synthetic datasets, provide critical ingredients for mathematical proofs, assist in proof writing, or assist with translation. The SfMComplete partial observations were produced by a rendering and structure-from-motion pipeline. Supporting qualitative and thematic data analysis is not applicable to this work.

All AI-assisted work was reviewed by the authors. Code was read, executed, and validated; reported quantitative results came from author-launched and inspected runs, with no numbers produced or edited by generative AI. References were verified against original publications, and text was revised and checked for technical accuracy. We take responsibility for the final content of this work, including all AI-assisted text, claims, and artifacts.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or sensitive data, and we identify no application-specific risk arising from shape completion of rigid objects beyond the general dual-use considerations of 3D reconstruction. The ethical consideration specific to this submission is the release of a derived dataset. SfMComplete is built from ShapeNet (Chang et al., 2015): its rendered views, structure-from-motion partial observations, distance fields, and watertight ground-truth meshes are all derived from ShapeNet source models. We will therefore distribute it under terms compatible with the ShapeNet terms of use, requiring recipients to accept those terms before obtaining access, and we will release the rendering and reconstruction scripts so that the benchmark can be regenerated from an independently obtained copy of ShapeNet. The KITTI (Geiger et al., 2012) data used for the qualitative in-the-wild result in Fig. **??** is used for non-commercial research under its CC BY-NC-SA 3.0 licence. The release will be accompanied by documentation covering composition, collection, preprocessing, intended uses, distribution, and maintenance, following the datasheet template of Gebru et al. (2021). We have read the ICLR Code of Ethics and this submission adheres to it.

REPRODUCIBILITY STATEMENT

We have made the following efforts to support reproducibility. The model is specified in Sec. 3, with the backbone we build on recapped in Appx. A; the four-stage training order is stated in the overview of Sec. 3 and given stage by stage in Appx. B, and the complete SDF supervision objective with its per-region terms and weights is derived in Appx. C. Architecture sizes, optimiser settings, and the number of function evaluations used at inference are reported in the implementation details of Sec. 4.1, which also defines every metric, its evaluation resolution, and the train/validation/test splits used for each benchmark. The construction of SfMComplete — the rendering and structure-from-motion pipeline, the derivation of each partial input modality, and the multi-resolution ground truth — is summarised in Sec. 4.1 and given in full in Appx. D. Source code for training, evaluation, and dataset construction, together with the SfMComplete data and the scripts that regenerate it, will be released publicly.

REFERENCES

Emanuele Aiello, Diego Valsesia, and Enrico Magli. Cross-modal learning for image-guided point cloud shape completion. In *Proc. NeurIPS*, 2022.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Michael S. Albergo, Mark Goldstein, Nicholas M. Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. In *Proc. ICML*, 2024.

Matan Atzmon and Yaron Lipman. SALD: Sign agnostic learning with derivatives. In *Proc. ICLR*, 2021.

Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Nießner. Scan2CAD: Learning CAD model alignment in RGB-D scans. In *Proc. CVPR*, pp. 2614–2623, 2019.

Ilya Baran and Jovan Popović. Automatic rigging and animation of 3D characters. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3), 2007.

Clément Chadebec et al. LBM: Latent bridge matching for fast image-to-image translation. *arXiv preprint arXiv:2503.07535*, 2025.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3D shape variational auto-encoders. In *Proc. CVPR*, 2025.

Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. AnchorFormer: Point cloud completion from discriminative nodes. In *Proc. CVPR*, 2023.

Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G. Schwing, and Liang-Yan Gui. SDFusion: Multimodal 3D shape completion, reconstruction, and generation. In *Proc. CVPR*, 2023.

Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3D shape reconstruction and completion. In *Proc. CVPR*, pp. 6970–6981, 2020.

In Cho, Youngbeom Yoo, Subin Jeon, and Seon Joo Kim. Representing 3D shapes with 64 latent vectors for 3D diffusion models. In *Proc. ICCV*, 2025.

Ruihang Chu, Enze Xie, Shaoshuai Mo, Zhenguo Li, Matthias Nießner, Chi-Wing Fu, and Jiaya Jia. DiffComplete: Diffusion-based generative 3D shape completion. In *Proc. NeurIPS*, pp. 75951–75966, 2023.

Ruikai Cui, Weizhe Liu, Weixuan Sun, Senbo Wang, Taizhang Shang, Yang Li, Xibin Song, Han Yan, Zhennan Wu, Shenzhou Chen, et al. NeuSDFusion: A spatial-aware generative model for 3D shape completion, reconstruction, and generation. In *Proc. ECCV*, pp. 1–18, 2024.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. CVPR*, pp. 5828–5839, 2017a.

Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3D-encoder-predictor CNNs and shape synthesis. In *Proc. CVPR*, pp. 5868–5877, 2017b.

Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Automated creation of digital cousins for robust policy learning. In *Proc. CoRL*, 2024.

James Davis, Stephen R. Marschner, Matt Garr, and Marc Levoy. Filling holes in complex surfaces using volumetric diffusion. *Proc. International Symposium on 3D Data Processing Visualization and Transmission*, 2002.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *Transactions on Machine Learning Research*, 2023.

Dror Freirich, Tomer Michaeli, and Ron Meir. A theory of the distortion-perception tradeoff in Wasserstein space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. arXiv:2107.02555.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64 (12):86–92, 2021.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proc. CVPR*, pp. 3354–3361, 2012.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proc. ICML*, pp. 3789–3799, 2020.

Ming Gui, Johannes Schusterbauer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. DepthFM: Fast monocular depth estimation with flow matching. In *Proc. AAAI*, 2025.

Ka-Hei Hui, Aditya Sanghi, Arianna Rampini, Kamal Rahimi Malekshan, Zhengzhe Liu, Hooman Shayani, and Chi-Wing Fu. Make-A-Shape: A ten-million-scale 3D shape model. In *Proc. ICML*, 2024.

Dequan Kong, Honghua Chen, Zhe Zhu, and Mingqiang Wei. BridgeShape: Latent diffusion schrödinger bridge for 3D shape completion. *arXiv preprint arXiv:2506.23205*, 2026.

Yangguang Li et al. TripoSG: High-fidelity 3D shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025a.

Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3D: Sparse representation and construction for high-resolution 3D shapes modeling. *arXiv preprint arXiv:2505.14521*, 2025b.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Proc. ICLR*, 2023.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos Theodorou, Weili Nie, and Anima Anandkumar. I^2SB: Image-to-image Schrödinger bridge. In *Proc. ICML*, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Proc. ICLR*, 2023b.

William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH*, 1987.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proc. CVPR*, pp. 4460–4470, 2019.

Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. AutoSDF: Shape priors for 3D completion, reconstruction and generation. In *Proc. CVPR*, pp. 306–315, 2022.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (SIGGRAPH)*, 41 (4), 2022.

Sanjeev Muralikrishnan, Vladimir G. Kim, Matthew Fisher, and Siddhartha Chaudhuri. Shape Unicode: A unified shape representation. In *Proc. CVPR*, 2019.

Guy Ohayon, Tomer Michaeli, and Michael Elad. Posterior-mean rectified flow: Towards minimum MSE photo-realistic image restoration. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.00418.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proc. CVPR*, pp. 165–174, 2019a.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proc. CVPR*, 2019b.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proc. ICCV*, pp. 4195–4205, 2023.

Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. ECCV*, pp. 523–540, 2020.

Yuchen Rao, Yinyu Nie, and Angela Dai. PatchComplete: Learning multi-resolution patch priors for 3D shape completion of unseen categories. In *Proc. NeurIPS*, pp. 34436–34450, 2022.

Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. XCube: Large-scale 3D generative modeling using sparse voxel hierarchies. In *Proc. CVPR*, 2024.

Simon Schaefer, Juan D. Galvis, Xingxing Zuo, and Stefan Leutenegger. SC-Diff: 3D shape completion with latent diffusion models. *arXiv preprint arXiv:2403.12470*, 2024.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger bridge matching. In *Proc. NeurIPS*, 2023.

Yawar Siddiqui, Duncan Frost, Samir Aroudj, Armen Avetisyan, Henry Howard-Jenkins, Daniel DeTone, Pierre Moulon, Qirui Wu, Zhengqin Li, Julian Straub, Richard Newcombe, and Jakob Engel. ShapeR: Robust conditional 3D shape generation from casual captures. *arXiv preprint arXiv:2601.11514*, 2026.

Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *Proc. CVPR*, 2021.

Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *Proc. ICCV*, 2017.

Maxim Tatarchenko, Stephan R. Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3D reconstruction networks learn? In *Proc. CVPR*, pp. 3405–3414, 2019.

Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *Proc. IROS*, 2017.

Bram Wallace and Bharath Hariharan. Few-shot generalization for single-image 3D reconstruction via priors. In *Proc. ICCV*, pp. 3818–3827, 2019.

Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive O-CNN: A patch-based deep representation of 3D shapes. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 37(6), 2018.

Francis Williams. Point cloud utils. `https://github.com/fwilliams/point-cloud-utils`, 2020.

Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *ECCV*, 2020.

Jianfeng Xiang et al. Structured 3D latents for scalable and versatile 3D generation. *arXiv preprint arXiv:2412.01506*, 2024.

Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP: Learning a unified representation of language, images, and point clouds for 3D understanding. In *Proc. CVPR*, 2023.

Xingguang Yan, Liqiang Lin, Niloy J. Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. ShapeFormer: Transformer-based shape completion via sparse representation. In *Proc. CVPR*, pp. 6239–6249, 2022.

Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. doi: 10.3390/s18103337.

Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PoinTr: Diverse point cloud completion with geometry-aware transformers. In *Proc. ICCV*, 2021.

Xumin Yu, Yongming Rao, Ziyi Wang, Jiwen Lu, and Jie Zhou. AdaPoinTr: Diverse point cloud completion with adaptive geometry-aware transformers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12):14114–14130, 2023.

Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point completion network. In *3DV*, 2018.

Faezeh Zakeri, Raphael Braun, Lukas Ruppert, and Hendrik P. A. Lensch. POC-SLT: Partial object completion with SDF latent transformers. In *Proc. Conference on Robots and Vision (CRV)*, 2025.

Biao Zhang, Matthias Nießner, and Peter Wonka. 3DILG: Irregular latent grids for 3D generative modeling. In *Proc. NeurIPS*, 2022a.

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.

Dongsu Zhang, Changwoon Choi, Inbum Park, and Young Min Kim. Probabilistic implicit scene completion. In *Proc. ICLR*, 2022b.

Lirui Zhang, Zhengkai Zhao, Zhi Zuo, Pan Gao, and Jie Qin. Simba: Towards high-fidelity and geometrically-consistent point cloud completion via transformation diffusion. In *Proc. AAAI*, 2026.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. CLAY: A controllable large-scale generative model for creating high-quality 3D assets. *ACM Transactions on Graphics*, 43(4), 2024.

Xuancheng Zhang, Yutong Feng, Siqi Li, Changqing Zou, Hai Wan, Xibin Zhao, Yandong Guo, and Yue Gao. View-guided point cloud completion. In *Proc. CVPR*, 2021.

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.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, et al. Hunyuan3D 2.0: Scaling diffusion models for high resolution textured 3D assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

Xin-Yang Zheng, Yang Liu, Peng-Shuai Wang, and Xin Tong. SDF-StyleGAN: Implicit SDF-based StyleGAN for 3D shape generation. *Computer Graphics Forum*, 41(5):52–63, 2022.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *Proc. ICLR*, 2024.

A BACKBONE VAE: COD-VAE

UniComplete builds on COD-VAE Cho et al. (2025), a variational autoencoder we adopt for two design properties that carry directly into the efficient high-resolution setting: a much more compact set latent, and a triplane plus shallow-MLP field decoder that scales to high output resolution, both improvements over the rest of the 3DShape2VecSet family Zhang et al. (2023); Zhao et al. (2023); Zhang et al. (2024); Li et al. (2025a); Zhao et al. (2025); Chen et al. (2025). We recap below the components our method depends on; the deviations introduced for shape completion are stated where they arise.

Encoder. COD-VAE encoder takes a point cloud $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{N}$ sampled from an object surface and outputs an unordered set of latents $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^{M}$ with $\mathbf{z}_i \in \mathbb{R}^D$ and $M \ll N$. To bridge the large compression gap, the encoder departs from 3DShape2VecSet's direct point-to-latent cross-attention Zhang et al. (2023) by inserting an intermediate patch-level representation: stacked cross- and self-attention blocks progressively transfer information from high-resolution point features to intermediate patches and finally to the M compact latents, all carried at a working feature width C (with $D \ll C$). A final KL block compresses the channel width from C to D and regularises $\mathcal{Z}$ toward a unit-variance Gaussian prior. The latent positions are anchored by farthest-point sampling (FPS) of the input points, an input-dependent choice that yields strong reconstruction but, as we discuss in Sec. 3.1, leaves the set without a consistent cross-input ordering.

Decoder. The latent decoder first restores the channel width of $\mathcal{Z}$ to C, then reconstructs triplane features at resolution R from the latent set: learnable triplane tokens act as mask tokens and are populated by transformer blocks reading from $\mathcal{Z}$, avoiding the cost of decoding a neural field directly from M vectors. An *uncertainty-guided token pruning* head discards triplane tokens flagged as low-uncertainty so decoder computation concentrates on geometrically detailed regions. For each query point $\mathbf{q} \in \mathbb{R}^3$, features are bilinearly sampled from the three planes, summed, and passed through a shallow MLP to predict occupancy, cheap enough to scale to high-resolution grids.

B TRAINING SCHEDULE

UniComplete is trained in four stages, each on top of the frozen output of the previous one; the losses are defined in the method subsections and only invoked here.

Stage 1: SDF VAE. The surface encoder for voxel grids, the anchored encoder, and the triplane–SDF decoder are trained on complete SDF grids with the surface-concentrated supervision of Eq. (4), in COD-VAE's two steps: the autoencoder first, then the KL block and latent decoder with the unit-Gaussian KL objective while the rest is frozen. The result is the SDF teacher and the only decoder in the system.

Stage 2: modality and partial encoders. With the SDF VAE frozen, each further modality encoder E^m (point cloud, UDF, occupancy, depth) is trained with Eq. (1) against the teacher's latent of the same complete shape. Each partial encoder E^m_{p} is then initialised from E^m and fine-tuned on partial–complete pairs with the same loss against the teacher's latent of the complete shape, which makes it the posterior-mean estimator of Sec. 3.3.

Stage 3: Surface Gate. The gate head is trained on the frozen triplane features of the SDF VAE as a binary classifier of surface cells (Sec. 3.4); the threshold is set after training.

Stage 4: flow. The velocity network v_θ is trained with Eq. (3) on pairs $(\mathbf{z}_{\mathrm{p}}, \mathbf{z}_{\mathrm{c}})$ produced by the frozen SDF partial encoder and the frozen SDF teacher; nothing else is updated. Because every partial encoder of Stage 2 emits its estimate on the same anchors, this one flow serves all modalities at inference.

C SDF SUPERVISION: LOSS DETAILS

This appendix gives the per-term formulae of the surface-concentrated supervision of Sec. 3.4. For each shape, a fixed budget of $|\mathcal{Q}|$ query points is drawn from three regions in fixed proportions: points on the surface, $\mathcal{S}$ (s_i=0); points in the near-surface band, $\mathcal{B}$ ($0 < |s_i| < \tau_{\mathrm{band}}$), balanced

inside and outside; and points in the outer region, $\mathcal{O}$ ($|s_i| \geq \tau_{\text{band}}$). Each term below is a mean over one of these sets, so its scale does not depend on the budget or on V. We write s_i for the truncated ground-truth distance at query i and $\hat{s}_i$ for the prediction.

Per-region ℓ_1 reconstruction. The primary term regresses the predicted distance to the truncated ground truth (Park et al., 2019a), with one weight per region so that the surface, the band, and the outer region can be balanced independently of how many points each receives:

$$\mathcal{L}_{\text{recon}} = \sum_{r \in \{\mathcal{S},\mathcal{B},\mathcal{O}\}} \frac{\lambda_r}{|r|} \sum_{i \in r} |\hat{s}_i - s_i|. \tag{5}$$

On the surface the target is zero, so the term reduces to $|\hat{s}_i|$ and acts as the boundary constraint of IGR (Gropp et al., 2020).

Sign-aware terms. Near the surface the distances are small, so an ℓ_1 error of a fraction of a voxel is cheap even when it flips the sign, and a flipped sign changes the extracted mesh. We therefore add inside–outside classification on the *same* predicted distance, without a separate occupancy head. In the band, a binary cross-entropy on a sharp sigmoid of $\hat{s}_i$ with temperature T_σ (in distance units) and ground-truth occupancy $y_i = \mathbf{1}[s_i < 0]$:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|\mathcal{B}'|} \sum_{i \in \mathcal{B}'} w_i \big[\, y_i \log \sigma(-\hat{s}_i/T_\sigma) + (1-y_i) \log\big(1 - \sigma(-\hat{s}_i/T_\sigma)\big) \,\big], \tag{6}$$

where $\mathcal{B}' \subset \mathcal{B}$ excludes points with $|s_i| < \epsilon$, whose label is ambiguous at the zero level set, and w_i balances the two classes and ramps up with $|s_i|$ so that the term does not thicken the surface. In the outer region the sign is easy, since $|s_i|$ is large, so a hinge that asks the prediction to be on the correct side by a margin of δ suffices:

$$\mathcal{L}_{\text{hinge}} = \frac{1}{|\mathcal{O}|} \sum_{i \in \mathcal{O}} w_i \left[\delta - \operatorname{sgn}(s_i)\, \hat{s}_i\right]_+. \tag{7}$$

Together the two terms give the single SDF head the inside–outside supervision of an occupancy network on top of the continuous regression of DeepSDF.

Surface gradient regularisers. At a subset $\mathcal{S}_g \subset \mathcal{S}$ of surface points we estimate $\nabla \hat{s}_i$ by central finite differences over six offset queries and apply two regularisers (Gropp et al., 2020; Atzmon & Lipman, 2021): alignment of the gradient direction with the ground-truth mesh normal $\mathbf{n}_i$,

$$\mathcal{L}_{\text{normal}} = \frac{1}{|\mathcal{S}_g|} \sum_{i \in \mathcal{S}_g} \Big(1 - \big\langle \nabla \hat{s}_i / \|\nabla \hat{s}_i\|,\ \mathbf{n}_i / \|\mathbf{n}_i\| \big\rangle\Big), \tag{8}$$

and a clamped eikonal residual that keeps the gradient magnitude near one,

$$\mathcal{L}_{\text{eik}} = \frac{1}{|\mathcal{S}_g|} \sum_{i \in \mathcal{S}_g} \operatorname{clip}\big(\|\nabla \hat{s}_i\| - 1,\ \pm\delta_\nabla\big)^2. \tag{9}$$

Finite differences cost six extra MLP queries per point but avoid the second-order autodiff graph of an analytic eikonal loss. Both regularisers are switched on after a short warm-up and ramped in with a cosine schedule, since they destabilise training when applied from the start.

Total loss. The five terms are combined with the weights of Eq. (4) in the main text, $\lambda_{\text{r}}, \lambda_{\text{b}}, \lambda_{\text{h}}, \lambda_{\text{n}}, \lambda_{\text{e}}$, and supervise both the autoencoder and the KL stages of the SDF VAE (Appx. B).

D SfMComplete: Construction and Statistics

This appendix gives the construction of SfMComplete summarised in Sec. 4.1: how the partials are produced, how the five partial modalities and the ground truth are derived, how the data is split, and how the partials differ from synthetic scans.

Objects and camera trajectories. We start from ShapeNet (Chang et al., 2015), use the eight categories of the 3D-EPN convention (Dai et al., 2017b), about 35K objects, and normalise every mesh to the unit cube. We simulate an agent driving past the object with a fixed, forward-facing camera. Each object is observed along four trajectories parallel to its longer axis, one on each side traversed in both directions, and the visible coverage of a trajectory is capped at 180° of the object, so no single partial sees the whole shape.

SfM reconstruction. A feature-based SfM pipeline reconstructs a sparse point cloud from each rendered trajectory, which gives four partials per object and about 140K partials in total. The reconstruction fails where real pipelines fail: textureless regions leave holes, density follows the viewpoint, pose drift smears surfaces, and mismatched features leave floaters; these are the artefacts that synthetic depth scanning cannot produce and that a completion model deployed after SfM must handle.

Partial modalities. From each partial point cloud we derive the grid modalities over the normalised $[-1, 1]^3$ domain with `point_cloud_utils` (Williams, 2020; Davis et al., 2002): the unsigned distance field directly, the signed distance field after a watertighting step that closes the open partial surface, and the occupancy grid as the sign of that field. Depth maps are rendered directly from the CAD mesh, with one camera at each of four corners of the object's bounding box looking at its center, which gives four depth partials per object. All grid partials are provided at $V \in \{64, 128, 256, 512\}^3$.

Ground truth. Signed distance fields and watertight meshes are computed directly from the normalised CAD mesh at the same four resolutions, so every partial is paired with the exact shape it was rendered from and with the same ground truth at every resolution.

Splits and statistics. Each category is split $80/5/15$ by object identity, following 3D-EPN; the four trajectories of an object share a split, so no test object is seen in training from another side.

Comparison to other datasets

Table 9: SfMComplete against existing partial-shape benchmarks. PC: point cloud; occ.: occupancy.

Benchmark	Partial source	Partial modalities	Ground truth	Pipeline noise
3D-EPN (Dai et al., 2017b)	synthetic depth	TSDF	32^3–128^3 TUDF, CAD mesh	$\times$
PatchComplete (Rao et al., 2022)	synthetic depth / ScanNet	TSDF	32^3 TSDF, Scan2CAD alignment	$\times$ / ✓
ShapeR (Siddiqui et al., 2026)	VI-SLAM in the wild	PC	scene annotation	✓
SfMComplete (ours)	SfM on rendered views	TSDF, UDF, occ., PC, depth	64^3–512^3 TSDF, CAD mesh	✓

Comparison with synthetic scans. Against virtual depth scanning, the SfM partials have lower and viewpoint-dependent point density, lower and more variable coverage, and points off the true surface. Against benchmarks captured in the wild, SfMComplete keeps exact ground truth: PatchComplete's real split (Rao et al., 2022) aligns ScanNet objects to CAD models through Scan2CAD (Avetisyan et al., 2019) at 32^3, and ShapeR (Siddiqui et al., 2026) releases 178 objects captured with visual-inertial SLAM and scene-level annotations without per-object CAD. SfMComplete has two orders of magnitude more objects than ShapeR, and pairs each partial with its CAD model by construction.

Examples. Fig. 4 shows the same SfMComplete partial in all five modalities next to the partial inputs of 3D-EPN (Dai et al., 2017b) and PatchComplete (Rao et al., 2022) for the same object, together with the ground truth, across categories. Compared with the synthetic scans of the two benchmarks, the SfM partials exhibit holes in textureless regions and viewpoint-dependent density.

Release. We will release the partial inputs in all modalities, the multi-resolution SDF and mesh ground truth, the splits, and the rendering-and-SfM construction pipeline under an anonymised URL; the pipeline extends to further categories and other SfM tools.

E Benchmarks, Metrics, and Implementation Details

3D-EPN. The benchmark (Dai et al., 2017b) contains 25,590 training and 5,384 test instances across eight ShapeNet categories. Each instance pairs six virtually scanned 32^3 truncated signed distance field (TSDF) partials with the complete shape as a truncated unsigned distance field (TUDF) at 32^3,

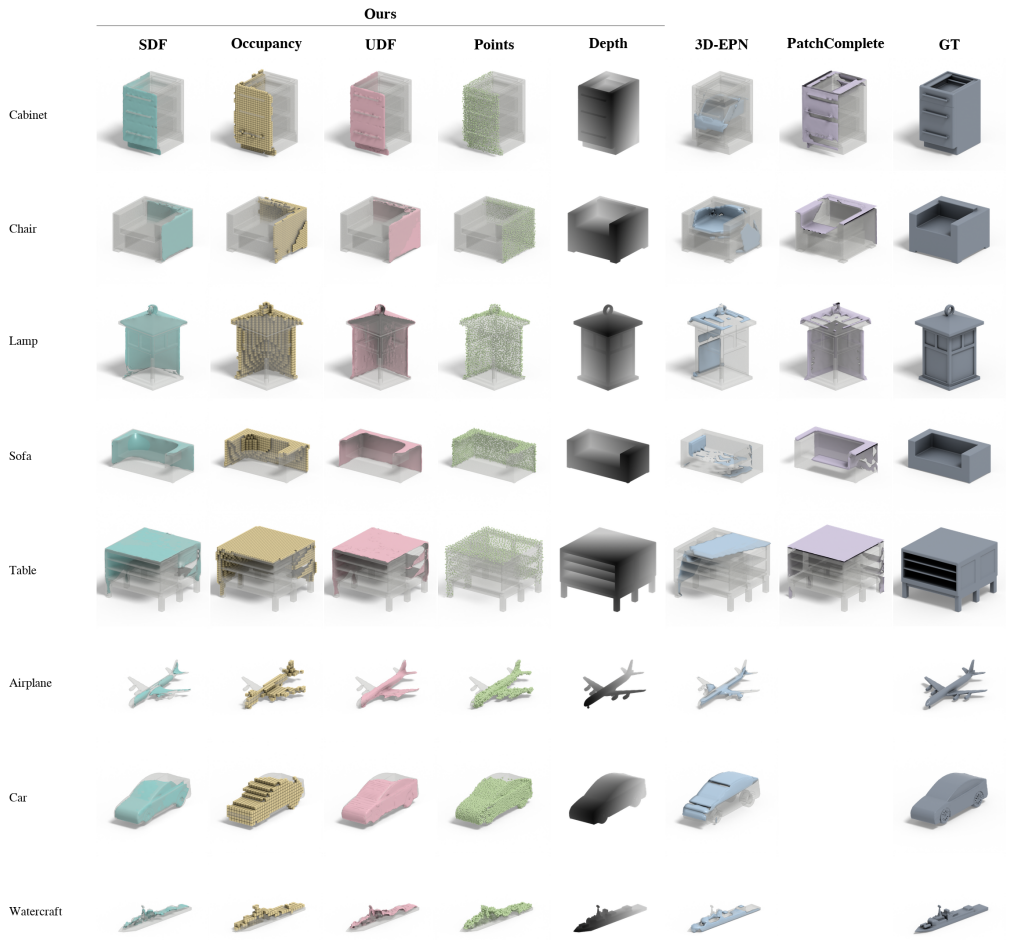


Figure 4: **SfMComplete examples.** One object per row: the same partial scan in five modalities (SDF, occupancy, UDF, point cloud, depth), the partial inputs of 3D-EPN and PatchComplete, and the ground truth. PatchComplete covers five of our eight categories, hence the three empty rows.

64^3, or 128^3. Completion is scored by the mean ℓ_1 error between predicted and ground-truth TUDF values averaged over all voxels.

PatchComplete. The benchmark (Rao et al., 2022) evaluates generalisation to unseen categories on a synthetic ShapeNet split (18 training and 8 test categories, 3,202 and 1,325 shapes, four partial scans per model) and a real split cropped from ScanNet (Dai et al., 2017a) scans with complete shapes from Scan2CAD (Avetisyan et al., 2019); all shapes are 32^3 TSDFs. We follow its metrics: the ℓ_1 Chamfer Distance on 10K surface samples per mesh, and the volumetric IoU.

SfMComplete. The train, validation, and test splits keep 80%, 5%, and 15% of each class by object identity (Appx. D). We report the ℓ_1 Chamfer Distance and volumetric IoU as above, and the F-Score at threshold τ=0.01 (Tatarchenko et al., 2019) on the normalised $[-1, 1]^3$ domain.

Baseline protocol. Every baseline is trained on the SfMComplete training split as one model for all eight categories and evaluated on all 21,340 test partial–complete pairs (5,353 objects, up to four trajectories each). SDF methods are meshed by marching cubes at level 0 on their native grid, and point methods by screened Poisson reconstruction (depth 9, 5% density trim). CD and F1 compare 10K points sampled from each mesh with a fixed set of 10K ground-truth surface points, summing both CD directions; IoU is computed from $\mathrm{sdf}\leq 0$ on the native grid for SDF methods and from the winding number of the Poisson mesh at 64^3 for point methods. Generative methods draw K=5 completions per partial. Each (partial, k) pair has a fixed seed shared by all methods that drives both

Table 10: Quantitative results of shape completion on 3D-EPN.

Method	Chair	Table	Sofa	Lamp	Plane	Car	Dresser	Watercraft	Avg.
3D-EPN Dai et al. (2017b)	0.418	0.377	0.392	0.388	0.421	0.259	0.381	0.356	0.374
ConvONet Peng et al. (2020)	0.210	0.247	0.254	0.234	0.185	0.195	0.250	0.184	0.220
SDF-StyleGAN Zheng et al. (2022)	0.321	0.256	0.289	0.280	0.295	0.224	0.273	0.282	0.278
cGCA Zhang et al. (2022b)	0.174	0.212	0.179	0.239	0.170	0.161	0.204	0.143	0.185
AutoSDF Mittal et al. (2022)	0.201	0.258	0.226	0.275	0.184	0.187	0.248	0.157	0.217
ShapeFormer Yan et al. (2022)	0.104	0.175	0.133	0.176	0.136	0.127	0.157	0.119	0.141
PatchComplete Rao et al. (2022)	0.134	0.095	0.084	0.087	0.061	0.053	0.134	0.058	0.088
DiffComplete Chu et al. (2023)	0.070	0.073	0.061	0.059	0.015	0.025	0.086	0.031	0.053
BridgeShape Kong et al. (2026)	0.055	0.059	0.047	0.038	0.012	0.023	0.055	0.022	0.039
UniComplete (Ours)	0.063	0.054	0.050	0.046	0.015	0.028	0.065	0.023	0.043

the sampler noise and the surface sampling, so every completion is reproducible. We report the best and the mean over the K draws for each metric, and the total mutual difference (TMD) for their spread; deterministic methods have one output.

Baseline training. We run the official code of each baseline and change only what our data requires; checkpoints are selected on held-out data. *DiffComplete* (Chu et al., 2023) (3D U-Net denoiser with a ControlNet branch, 43M parameters) is trained at 64^3 as a DDPM (T=1000, linear schedule, ϵ-prediction) with Adam at 10^{-4}, batch size 32 and mixed precision, with the partial encoded as $(|s|, \operatorname{sgn} s)$ as in the original. The target is the signed SDF centred by its training mean rather than the unsigned TUDF, since IoU needs the sign. We sample with the paper's 100-step respacing; since the ϵ-model does not control the global level of the field, at each reverse step we shift the x_0 prediction so that its 0.75-quantile equals the truncation value. *BridgeShape* (Kong et al., 2026) is trained at 64^3 and 128^3 with signed SDF targets and partial inputs at the output resolution. A VQ-VAE maps the grid to a 3×16^3 latent with 8,192 codes (the 128^3 model adds one stride-2 level), and a latent Schrödinger bridge (T=1000) sampled in 3 steps maps the partial latent to the complete one. The bridge is trained with AdamW at 2×10^{-4} for about 200K steps at both resolutions; the VQ-VAE is trained with Adam at 10^{-4}. *NeuSDFusion* (Cui et al., 2024) is conditioned on the partial point cloud: a triplane VAE ($3\times32\times256^2$ triplanes, $4\times16\times48$ latent) with an SDF MLP is trained on 256^3 SDFs, and a latent diffusion model (T=1000, cosine schedule, x_0-prediction) reads the cloud through a point encoder with cross-attention. We sample with DDIM (η=0, 5 steps; 3 to 100 steps gave the same IoU in a probe) on 1,024 random input points per completion, keep one raw draw instead of the upstream best-of-8 consistency filter, and disable condition dropout. *AdaPoinTr* (Yu et al., 2023) is trained from scratch with the official recipe (batch 16, 600 epochs), and *Simba* (Zhang et al., 2026) is fine-tuned from its released PCN weights for 100 epochs (learning rate 10^{-4}, 5 warm-up epochs) with inputs rescaled to the PCN frame; both map 2,048 partial points to 16,384 points.

Implementation details. The anchored VAE uses M=40 canonical anchors, latent dimension D=32, and working width C=512, keeping COD-VAE's encoder and decoder configuration with the triplane resolution matched to the output SDF grid. The surface encoder centres each kernel-3 sparse convolution on the surface-band voxels, giving a three-voxel receptive band along each axis. The flow network is an 11-layer TripoSG-variant diffusion transformer (Li et al., 2025a; Peebles & Xie, 2023). All stages are trained with Adam at learning rate 10^{-4} in FP16 mixed precision on 8×NVIDIA RTX A6000 GPUs with 48 GB each. The autoencoder and KL stages of the SDF VAE are trained for 200 and 280 epochs at batch sizes 64 and 128, and the flow for 100K iterations at batch size 128.

F Experiments and Ablations

F.1 Per-category Results on 3D-EPN and PatchComplete

The main text (Sec. 4.2) reports category averages; Tab. 10 gives the per-category ℓ_1 on 3D-EPN, and Tab. 11 and Tab. 12 the per-category CD and IoU on the synthetic and real-world splits of PatchComplete.

Table 11: Shape completion results on PatchComplete *synthetic* split for *unseen* categories.

Method	Bag		Lamp		Bathtub		Bed		Basket		Printer		Laptop		Bench		Avg.	
	CD↓	IoU↑	CD	IoU	CD	IoU	CD	IoU	CD	IoU	CD	IoU	CD	IoU	CD	IoU	CD	IoU
3D-EPN Dai et al. (2017b)	5.01	73.8	8.07	47.2	4.21	57.9	5.84	58.4	7.90	54.0	5.15	73.6	3.90	62.0	4.54	48.3	5.58	59.4
Few-Shot Wallace & Hariharan (2019)	8.00	56.1	15.1	25.4	7.05	45.7	10.0	39.6	8.72	40.6	9.26	56.7	10.4	31.3	8.11	27.2	9.58	40.3
IF-Nets Chibane et al. (2020)	4.77	69.8	5.70	50.8	4.72	55.0	5.34	60.7	4.44	50.2	5.83	70.5	6.47	58.3	5.03	49.7	5.29	58.1
Auto-SDF Mittal et al. (2022)	5.81	56.3	6.57	39.1	5.17	41.0	6.01	44.6	6.70	39.8	7.52	49.9	4.81	51.1	4.31	39.5	5.86	45.2
ConvONet Peng et al. (2020)	5.10	70.8	5.42	52.6	4.96	60.4	5.42	63.2	6.16	54.6	5.56	72.1	4.78	57.3	4.69	49.6	5.26	60.1
PatchComplete Rao et al. (2022)	3.94	77.6	4.68	56.4	3.78	66.3	4.49	66.8	5.15	61.0	4.63	77.6	3.77	63.8	3.70	53.9	4.27	65.4
DiffComplete Chu et al. (2023)	3.86	78.3	4.80	57.9	3.52	68.9	4.16	67.1	4.94	65.5	4.40	76.8	3.52	67.4	3.56	58.2	4.10	67.5
BridgeShape Kong et al. (2026)	3.70	80.0	4.94	62.7	3.41	69.8	4.32	69.6	5.50	65.1	4.09	81.4	3.13	72.7	3.38	59.1	4.06	70.1
UniComplete (Ours)	3.83	80.6	5.21	56.5	3.84	66.9	4.43	67.0	4.39	63.4	4.23	79.2	3.26	70.2	3.65	57.5	4.10	67.7



Figure 5: Reconstruction of complete test shapes by the first-stage autoencoders: ours, BridgeShape's VQ-VAE (Kong et al., 2026), and NeuSDFusion's triplane VAE (Cui et al., 2024), each at its native resolution, against the ground truth.

Table 12: Shape completion results on PatchComplete *real-world* split for *unseen* categories.

Method	Bag		Lamp		Bathtub		Bed		Basket		Printer		Avg.	
	CD↓	IoU↑	CD	IoU	CD	IoU	CD	IoU	CD	IoU	CD	IoU	CD	IoU
3D-EPN Dai et al. (2017b)	8.83	53.7	14.3	20.7	7.56	41.0	7.76	47.8	7.74	36.5	8.36	63.0	9.09	44.0
Few-Shot Wallace & Hariharan (2019)	9.10	44.9	11.9	19.6	7.77	38.2	9.07	34.9	8.02	34.3	8.30	62.2	9.02	38.6
IF-Nets Chibane et al. (2020)	8.96	44.2	10.2	24.9	7.19	39.5	8.24	44.9	6.74	42.7	8.28	60.7	8.26	42.6
Auto-SDF Mittal et al. (2022)	9.30	48.7	11.2	24.4	7.84	36.6	7.91	38.0	7.54	36.1	9.66	49.9	8.90	38.9
ConvONet Peng et al. (2020)	9.12	52.5	9.83	20.3	7.93	41.2	8.14	41.6	7.39	37.0	7.62	64.9	8.34	42.9
PatchComplete Rao et al. (2022)	8.23	58.3	9.42	28.4	6.77	48.0	7.24	48.4	6.60	45.5	6.84	70.5	7.52	49.8
DiffComplete Chu et al. (2023)	7.05	48.5	6.84	30.5	8.22	48.5	7.20	46.6	7.42	59.2	6.36	74.5	7.18	51.3
BridgeShape Kong et al. (2026)	7.67	61.2	8.07	36.3	6.28	50.9	6.87	51.3	6.20	50.2	6.83	71.0	6.99	53.5
UniComplete (Ours)	7.32	52.8	8.87	31.8	6.55	49.2	6.93	50.4	6.06	54.4	6.35	73.6	7.01	52.0

F.2 VAE Reconstruction

The first-stage autoencoder sets an upper bound on the completion quality of a latent method. Fig. 5 compares reconstructions of complete test shapes by our SDF VAE and by the autoencoders of the latent baselines (Rec. IoU in Tab. 3). Our VAE keeps its fidelity as the resolution grows, whereas BridgeShape's VQ-VAE falls well behind ours at 128^3 and NeuSDFusion's triplane VAE stays well below ours at 256^3.

F.3 Completion Diversity

Fig. 6 shows several completions drawn from the same partial with different seeds; TMD in Tab. 3 measures this spread over the test set. Samples agree on the scanned surface and differ where nothing was observed—the cabinet front, the leg and apron structure under the seats and table tops.

F.4 Inference Time and Memory Efficiency

As requested, Tab. 13 reports the inference time and memory efficiency of our method against the SOTA baselines. SurfaceGate's avoidance of empty space and one-step transport on a compact latent both reduce time; sparsity reduces memory.

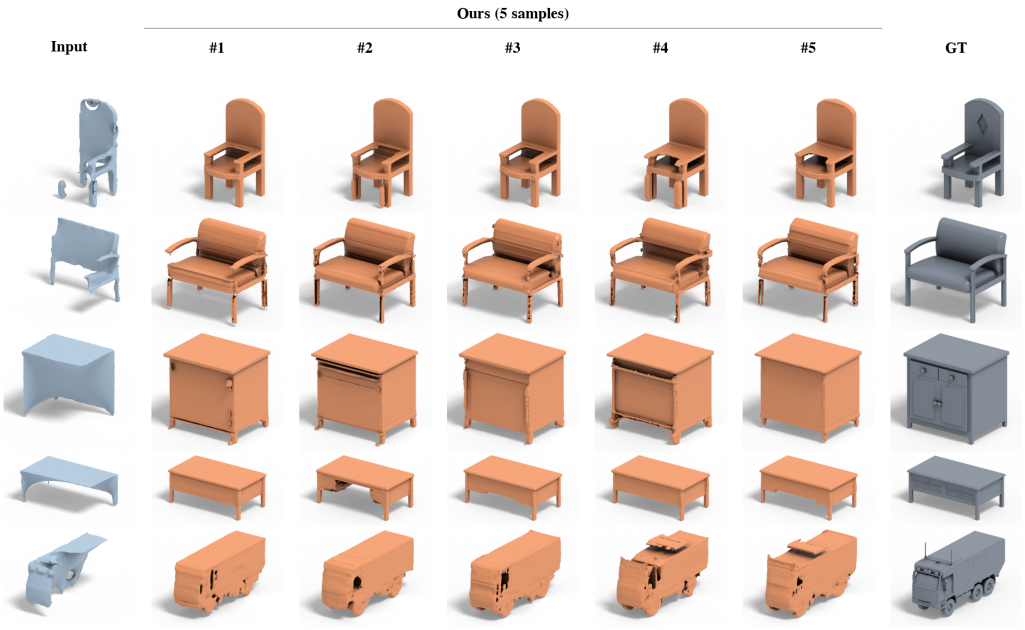


Figure 6: Completion diversity. Each row shows one partial input, several completions drawn with different seeds, and the ground truth.

Protocol. All measurements are taken on a single 48 GB NVIDIA A40 at batch size 1. *Training memory* is the peak reserved GPU memory of one optimisation step in 16-bit mixed precision, measured after a warm-up step. As the sweep extends beyond the resolutions of SfMComplete, all methods are fed the same analytic hollow-sphere SDF, whose surface-voxel count ($\approx 5R^2$) and interior occupancy ($\approx 4.7\%$), the two quantities that determine memory, match those of the SfMComplete test set. To keep its 16^3 latent, BridgeShape (Kong et al., 2026) adds one stride-2 level per doubling of R, with channel widths capped at its 128^3 configuration; our encoder takes only the surface band as sparse input. *Inference* is timed in FP32 on 15 test objects after 3 warm-up objects, with each stage delimited by CUDA synchronisation. Meshing uses the same marching-cubes routine for all methods and includes the device-to-host transfer; Peak VRAM is the maximum over the 15 objects. All baselines use their official samplers, with 3 bridge steps for BridgeShape, 100 DDPM steps for DiffComplete (Chu et al., 2023), and 5 DDIM steps for NeuSDFusion (Cui et al., 2024). Where the official sampler re-encodes the condition at every step, the encoder cost is included in Transport, hence the missing Encoder entries of DiffComplete and NeuSDFusion.

Table 13: Inference efficiency with batch size as 1. Stage times in ms; Peak VRAM in GiB.

Res.	Method	Encoder	Transport	Decoder	Meshing	Total	Peak VRAM
64^3	Ours	22.8	15.7	11.7	12.7	63.0	1.37
	BridgeShape	11.4	37.7	15.6	11.4	76.1	0.76
	DiffComplete	–	15898	–	13.7	15912	2.31
128^3	Ours	24.4	15.4	17.9	61.1	118.8	1.52
	BridgeShape	35.2	40.2	94.0	58.3	227.7	4.00
256^3	Ours	28.3	19.2	80.1	401.1	528.7	1.87
	NeuSDFusion	–	148.1	582.1	418.9	1149.6	5.02
512^3	Ours	38.2	18.6	579.9	2695	3331.2	4.68

F.5 Number of Anchors

Tab. 14 varies the number of canonical anchors while holding the VAE backbone, the transport, and the schedule fixed. Completion peaks at 40 anchors and degrades slightly at both 32 and 64; we therefore use 40 throughout.

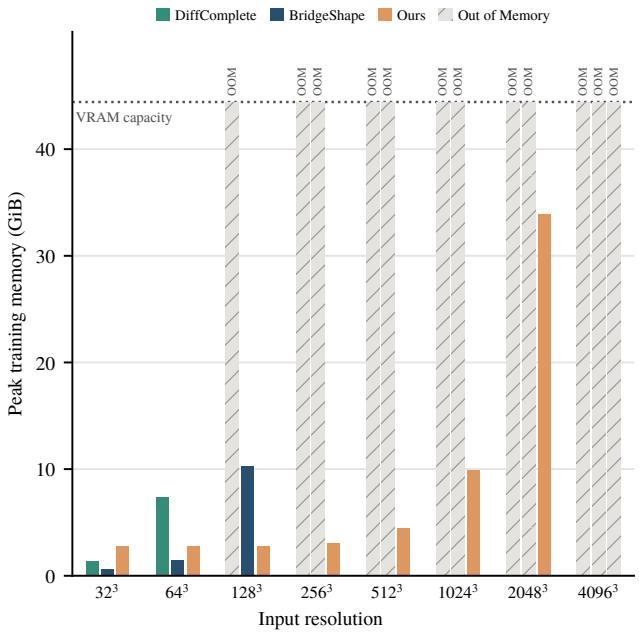


Figure 7: Peak training memory vs. input resolution (batch size 1, a single 48 GB NVIDIA A40). For DiffComplete (Chu et al., 2023), we report its full single-stage training; for BridgeShape (Kong et al., 2026) and ours, only the first-stage (VQ-)VAE. Hatched bars denote out-of-memory failures. The memory of the baselines' dense 3D CNNs scales as $\mathcal{O}(R^3)$. Our memory comprises a constant cost (parameters, optimizer states, fixed-size latent) and an $\mathcal{O}(R^2)$ surface term. The surface term remains small up to 256^3 and becomes dominant only beyond 512^3, which allows training at 2048^3.

Table 14: Completion on SfMComplete at 128^3 w.r.t. the number of anchors.

# Anchors	IoU ↑	CD ↓	$F1_{.01}$ ↑
32	62.53	4.72	86.12
40	63.31	4.51	87.06
64	62.98	4.64	86.57