S23DR 2026: End-to-End 3D Wireframe Prediction via DETR-Style Set Prediction with Contrastive Denoising

Nitiz Khanal
Pulchowk Campus, Lalitpur, Nepal

***Abstract*—We present *WireframeDETR*, our submission to the Structured Semantic 3D Reconstruction (S23DR) 2026 Challenge, which requires predicting a 3D building wireframe from multi-view COLMAP point clouds. Our method applies DETR [5]-style set prediction directly to 3D point clouds, producing wireframes as sets of edge coordinate pairs without any intermediate vertex detection stage. We introduce three technical contributions: (1) contrastive denoising training that stabilises noisy Hungarian matching in early epochs; (2) a multi-scale encoder that aggregates the last K encoder layer outputs via learned scalar weights; and (3) progressive auxiliary loss weighting that concentrates gradient signal on the decoder layers that most benefit from it. Our model achieves a public test HSS of 0.575 (F1 = 0.664, IoU = 0.516) and a best validation HSS of 0.534 on the cleaned val split.**

I. Introduction

The S23DR 2026 Challenge [1] requires predicting the 3D wireframe of a building rooftop — a set of 3D vertices and connecting edges — from a sparse COLMAP [8] point cloud augmented with Gestalt and ADE20K semantic labels and monocular depth estimates. Performance is measured by the Hybrid Structure Score (HSS), the harmonic mean of vertex F1 and edge volumetric IoU:

$$\text{HSS} = \text{harmonic\_mean}(F1(gt_v, \hat{v}),\, \text{IoU}(gt_e, \hat{e})). \quad (1)$$

We explored three directions before arriving at WireframeDETR.

Path A: Perceiver fine-tuning. The official learned baseline [2] is a Perceiver transformer (val HSS 0.350). Resuming training with stronger augmentation caused catastrophic forgetting: HSS fell below 0.26 within 20 000 steps.

Path B: Two-stage PointNet pipeline. Motivated by the 2025 winning solution [4], we trained a PointNet-style vertex detector (vertex F1 = 0.655, recall = 0.927) followed by an edge existence classifier (edge F1 = 0.289, precision = 0.169). The edge precision bottleneck proved difficult to close within the available compute budget; we set this direction aside.

Path C: WireframeDETR (final approach). We reformulate wireframe prediction as direct edge-set regression over the 3D point cloud. Each predicted edge is a 6D coordinate pair $(x_1, y_1, z_1, x_2, y_2, z_2)$; Hungarian matching assigns predictions to ground-truth edges at training time. This avoids cascaded error propagation and handles variable-cardinality wireframes without post-hoc NMS.

II. Related Work

3D wireframe and structured reconstruction. Classical methods recover building structure from point clouds via planar primitive fitting and intersection. PolyFit [10] selects a minimal planar face set via integer programming; City3D recovers LoD-2 models from airborne LiDAR using RANSAC roof planes. Both require explicit intermediate representations and handcrafted priors. The 2025 challenge winning solution [4] uses two PointNet-like networks: one refines 3D vertex candidates and a second classifies per-pair edge existence, achieving HSS = 0.43 on the private test set.

DETR-style set prediction. DETR [5] reformulates object detection as direct set prediction via bipartite matching, eliminating NMS and hand-designed anchors. DN-DETR [6] identifies noisy Hungarian assignments in early training as the chief convergence bottleneck and injects ground-truth-aligned denoising queries to stabilise gradients. DINO-DETR [7] further improves with contrastive denoising and mixed query initialisation. RoomFormer [11] applies DETR-style set prediction to indoor floor plan reconstruction, demonstrating that set-prediction transformers can replace multi-stage graph pipelines for structured geometry. We extend this paradigm to outdoor building wireframes operating on sparse 3D point clouds.

III. Dataset

The dataset (`usm3d/hoho22k_2026_trainval`) contains approximately 22 000 building scenes. Each scene provides a COLMAP sparse point cloud with per-point RGB colour, per-image Gestalt (73-class building-specific) and ADE20K (107-class) segmentation maps, and monocular depth estimates from Metric3Dv2 [9]. Ground-truth wireframes vary widely in complexity, from simple gable roofs to highly articulated multi-ridge structures.

Challenges. COLMAP reconstruction is sparse on textureless surfaces and frequently incomplete. The dataset also contains scenes with inconsistent camera parameters causing pose–annotation misalignment, where projected 3D vertices do not align with their expected Gestalt class pixels. Both issues complicate any approach that fuses the 3D point cloud with 2D segmentation information. Figure 1 illustrates the available input modalities.

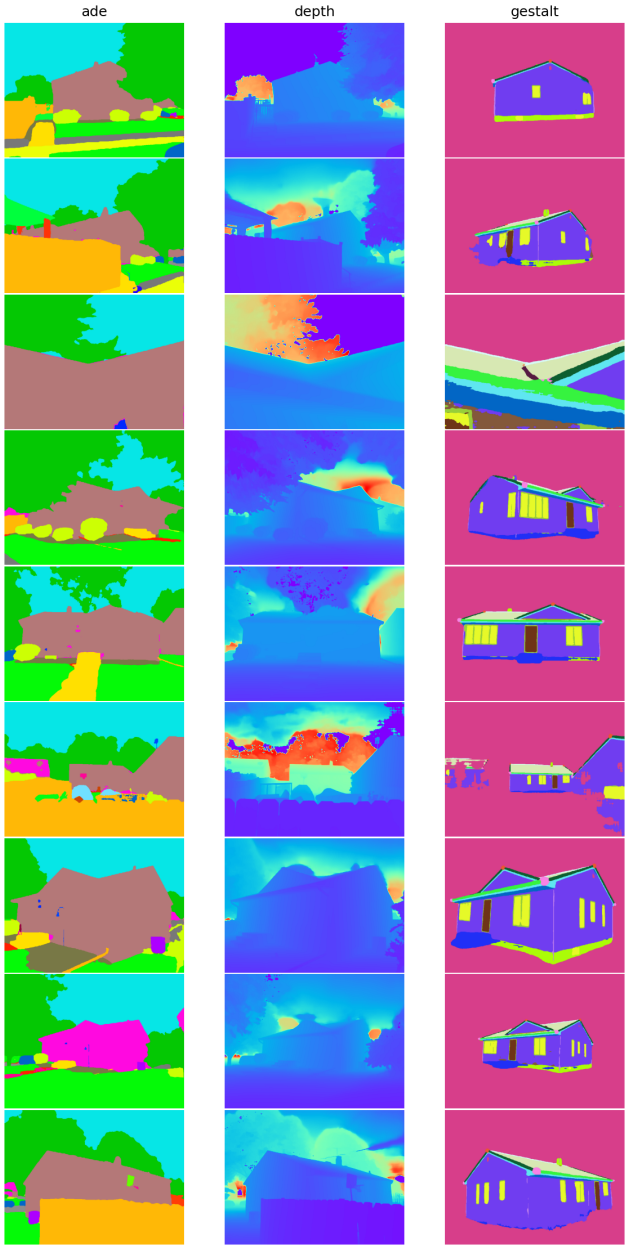

Fig. 1. Input modalities for a representative scene. Gestalt segmentation labels directly encode vertex and edge classes of the roof structure, which guides our sampling strategy.

Point cloud preprocessing. Each 3D COLMAP point is projected into all visible views; the most common Gestalt label across views is assigned to it [3]. Points whose dominant Gestalt label is *unclassified* or *unknown* are discarded. The final per-point feature is the 3-channel RGB colour, normalised to $[0, 1]$.

Gestalt-guided sampling. Gestalt labels identify structurally significant point classes: vertex classes (*apex*, *eave end point*, *flashing end point*) and edge classes (*ridge*, *rake*, *eave*, *hip*, etc.). Following [3], we assign sampling weights 100, 10, and 1 to vertex-class, edge-class, and background points

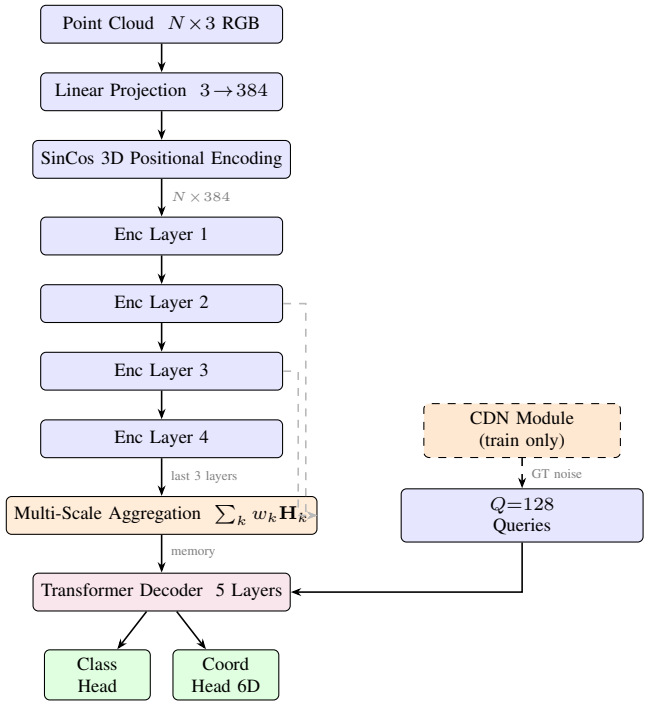

Fig. 2. WireframeDETR architecture. RGB features are projected to d=384, enriched with sinusoidal positional encoding, and processed by a 4-layer Transformer encoder. Multi-scale aggregation combines the last K=3 layer outputs via learned softmax weights (grey taps). At training time the CDN module injects denoising queries with known GT assignments; at inference only the Q=128 learned queries are used.

respectively, then draw N points proportional to these weights. This concentrates samples near structural features without encoding semantic labels as model features. We draw N=4,096 points during training and N=7,168 at inference.

Coordinate normalisation. Point coordinates are normalised to $[0, 1]^3$ per scene using the 5–95 % percentile range of each axis, providing a consistent scale across scenes of varying physical size.

Validation split. We evaluate on cleaned val: the official validation split filtered by a curated list of scenes with severe pose–annotation misalignment, retaining approximately 1 150 clean scenes for evaluation.

IV. Method

A. Architecture Overview

Figure 2 illustrates *WireframeDETR*. A linear projection maps the 3-dimensional RGB features to embedding dimension $d = 384$. A 3D sinusoidal positional encoding (SinCos3DPE) splits d evenly across XYZ axes and is added to each point token. A 4-layer multi-scale Transformer encoder processes all N point tokens. A 5-layer Transformer decoder with $Q = 128$ learned edge queries produces the final predictions.

At training time, Hungarian matching assigns predictions to ground-truth edges minimising a combined cost of L1 coordinate distance and negative classification probability. The

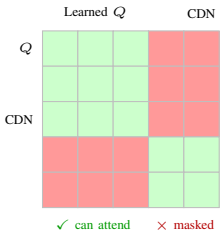

Fig. 3. CDN attention mask. Learned and denoising queries attend only within their own group, keeping gradient paths independent.

training loss has two terms: cross-entropy (weight 1.0) and L1 coordinate regression (weight 5.0). Auxiliary outputs from each intermediate decoder layer are supervised with progressively increasing weight (Section IV-D).

The model has approximately 22.7 M parameters.

B. Contrastive Denoising Training

Standard DETR training suffers from unstable Hungarian assignments in early epochs, where the same query is matched to different GT targets across iterations, producing contradictory gradients [6]. We address this with contrastive denoising (CDN) adapted from DN-DETR [6].

At each training step, for each of the M ground-truth edges we generate $G{=}5$ *positive* denoising queries by perturbing GT coordinates with noise $\epsilon \sim \mathcal{U}[-\lambda_+, \lambda_+]^6$, $\lambda_+ = 0.4$, and $G{=}5$ *negative* queries with larger noise $\lambda_- = 0.8$. Positive queries are supervised against their originating GT edge; negative queries are supervised as background. An attention mask (Figure 3) prevents cross-attention between denoising and learned queries, preserving independence.

The decoder sequence length at training time is $Q + 2GM$ (varying per scene as M differs). The CDN loss uses the same coefficients as the main loss. During training, the CDN coordinate loss decreases from ≈2.07 at epoch 0 to ≈1.25 at epoch 15, confirming that the denoising path converges steadily alongside the main queries.

C. Multi-Scale Encoder Memory

A standard Transformer encoder passes only the final layer's output to the decoder, discarding intermediate representations. We collect all encoder layer outputs $\{\mathbf{H}_1, \ldots, \mathbf{H}_L\}$ and compute a learned weighted sum of the last $K{=}3$ outputs:

$$\mathbf{M} = \sum_{k=1}^{K} \frac{e^{w_k}}{\sum_{j=1}^{K} e^{w_j}} \mathbf{H}_{L-K+k}, \tag{2}$$

where $w_1, w_2, w_3 \in \mathbb{R}$ are learned scalar weights initialised to 0. This gives the decoder access to both local geometry (earlier layers) and abstract structural features (final layer) at a cost of three extra parameters.

D. Progressive Auxiliary Loss Weighting

Auxiliary outputs after intermediate decoder layers carry steadily improving prediction quality; we reflect this by weighting the i-th auxiliary output ($i = 0, \ldots, N_{\text{aux}} - 1$) as:

$$w_i^{\text{aux}} = 0.5 + 0.5\,\frac{i+1}{N_{\text{aux}}}, \tag{3}$$

linearly increasing from 0.6 (first auxiliary) to 1.0 (pre-final layer).

E. Post-Processing and Inference

Predicted edges are filtered by classification score (> 0.9). The resulting vertex set is de-duplicated by iteratively merging vertices within 0.5 m, placing each merged vertex at the intersection of incident line directions [3]. We adopt this post-processing from publicly available code [3].

Optionally, four Y-axis rotations ($0°, 90°, 180°, 270°$) can be applied at inference time; predictions from each rotation are un-rotated and merged via the same vertex deduplication step. WireframeDETR treats points as an unordered set and is therefore compatible with rotational augmentation at inference. This is disabled by default as its contribution was not isolated in ablation.

F. Training Setup

We train with AdamW (lr=10^{-4}, weight decay 10^{-4}), a one-cycle LR schedule, batch size 14, and FP16 mixed precision on a single NVIDIA A100 80 GB GPU for 200 epochs (≈27 hours). Data augmentation: random Y-axis rotation and random horizontal flip. We disable `torch.compile` because CDN's variable decoder sequence length ($Q + 2GM$) triggers repeated Triton kernel recompilation. Gradient ℓ_2 norm is clipped at 0.1 to stabilise CDN training.

V. Experiments

A. Main Results

Table I summarises validation HSS across our development progression. WireframeDETR with plain 3-channel RGB features and Gestalt-guided sampling achieves a public test HSS of **0.575** and a best validation HSS of 0.534, substantially exceeding the official Perceiver baseline (0.350).

B. CDN Training Dynamics

Figure 4 shows the CDN coordinate loss over the first 15 epochs. The loss decreases steadily from 2.07 to 1.25, confirming that denoising queries — which have fixed GT assignments bypassing noisy Hungarian matching — converge reliably from the first iteration. CDN classification loss falls from 0.170 to 0.137 over the same window. Both are consistently lower than the corresponding main loss terms, as expected when queries receive cleaner gradient signal.

TABLE I

RESULTS ACROSS DEVELOPMENT STAGES. †PUBLIC LEADERBOARD (TEST SPLIT); VERT. F1 AND EDGE IOU FROM THE LEADERBOARD `corner_f1` AND `edge_iou` METRICS. ‡CLEANED VALIDATION SPLIT. PATH B LEADERBOARD SCORES REFLECT THE FULL TWO-STAGE SYSTEM; INTERNALLY THE VERTEX DETECTOR ACHIEVED F1 = 0.655 AND THE EDGE CLASSIFIER F1 = 0.289.

Approach	Split	Vert. F1	Edge IoU	HSS
Perceiver baseline [2]	Val‡	—	—	0.350
+ fine-tuning (Path A)	Val‡	—	—	<0.26
PointNet two-stage (Path B)	Test†	0.497	0.409	0.442
WireframeDETR (ours)	Val‡	0.603	0.471	0.534
WireframeDETR (ours, **best**)	Test†	**0.664**	**0.516**	**0.575**

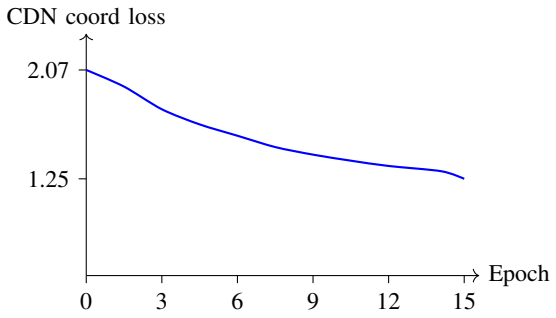

Fig. 4. CDN coordinate loss over the first 15 epochs. Steady convergence confirms the denoising path provides clean gradient signal from epoch 0.

C. Auxiliary Loss Ordering

At all observed training epochs the auxiliary losses satisfy $\ell_{\text{aux\_0}} < \ell_{\text{aux\_1}} < \ell_{\text{aux\_2}} < \ell_{\text{aux\_3}} < \ell_{\text{main}}$, confirming that later decoder layers produce progressively better predictions and that progressive weighting does not cause any layer to under-perform its predecessors.

D. Comparison with Baselines

On the public leaderboard, WireframeDETR (HSS = 0.575) outperforms both the handcrafted baseline (HSS = 0.391 [1]) and the official learned Perceiver baseline (HSS = 0.474 [2]) by a substantial margin.

E. Limitations

Training is $\approx 2\times$ slower than the Perceiver baseline ($\approx$8 min/epoch vs. 4 min/epoch). CDN's variable-length decoder sequence precludes `torch.compile`, adding $\approx$30% overhead. TTA (4 rotations) multiplies inference time by $4\times$ but requires no additional training; it is disabled by default in our submission.

VI. CONCLUSION

We presented WireframeDETR, a DETR-based end-to-end model for 3D building wireframe prediction that directly regresses edge coordinate pairs from a sparse RGB point cloud. Our three contributions — contrastive denoising training, multi-scale encoder memory, and progressive auxiliary loss weighting — combine to achieve a public test HSS of **0.575** and a best validation HSS of 0.534 on the S23DR 2026 cleaned val split. Post-processing routines are adopted from [3]. Future work includes adding vertex feature heads for feature-guided vertex merging and exploring joint vertex–edge prediction with contrastive vertex losses.

REFERENCES

[1] J. Langerman, D. Mishkin, and Y. Huang, "S23DR: Structured Semantic 3D Reconstruction Challenge 2026," https://huggingface.co/spaces/usm3d/S23DR2026, 2026.

[2] J. Langerman, "S23DR 2026 learned baseline (Perceiver)," https://huggingface.co/usm3d/learned-baseline-2026, 2026.

[3] jastermark, "S23DR2026: Open-source EdgeDETRPE implementation," https://huggingface.co/jastermark/S23DR2026, 2026.

[4] J. Skvrna and L. Neumann, "Structured Semantic 3D Reconstruction (S23DR) Challenge 2025 – Winning Solution," *arXiv preprint*, 2025.

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. ECCV*, 2020, pp. 213–229.

[6] F. Li, H. Zhang, L. Liu, J. Guo, L. Ni, and L. Zhang, "DN-DETR: Accelerate DETR training by introducing query denoising," in *Proc. CVPR*, 2022, pp. 13619–13627.

[7] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. Ni, and H.-Y. Shum, "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," in *Proc. ICLR*, 2023.

[8] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. CVPR*, 2016, pp. 4104–4113.

[9] M. Hu et al., "Metric3D v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation," *IEEE TPAMI*, 2024.

[10] L. Nan and P. Labatut, "PolyFit: Polygonal surface reconstruction from point clouds," in *Proc. ICCV*, 2017, pp. 2353–2361.

[11] J. Yue et al., "RoomFormer: Encoding floor plans as sequences of rooms," in *Proc. CVPR*, 2023, pp. 2121–2130.