Unified Panoramic Geometry Estimation via Multi-View Foundation Models

Vukasin Bozic ETH Zürich
Isidora Slavkovic Google
Dominik Narnhofer ETH Zürich
Nando Metzger ETH Zürich
Denis Rozumny Meta
Konrad Schindler ETH Zürich
Nikolai Kalischek Google

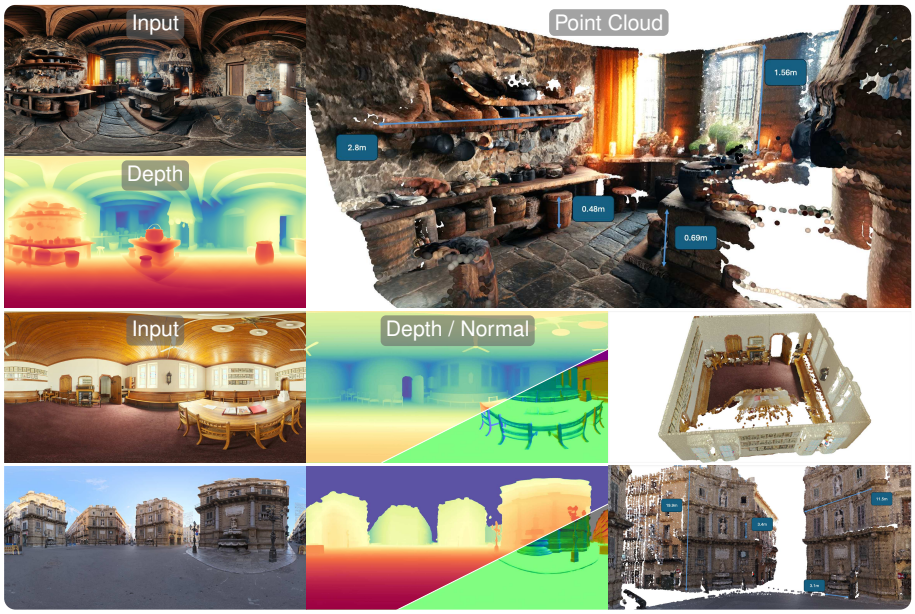

Figure 1: **Different geometric modalities predicted by PaGeR.** Given only a single monocular panoramic image, our framework simultaneously reconstructs highly detailed scale-invariant depth, absolute metric depth, surface normals, and sky segmentation masks across both indoor and outdoor environments. Spatial measurements are presented in meters.

Abstract

Geometry estimation from perspective images has greatly advanced, maturing to the point where off-the-shelf foundation models are able to reconstruct 3D scene structure not only from multi-view imagery, but even from a single view. A natural extension is 3D reconstruction from panoramas, with the exciting prospect of recovering a full 360° scene from a single panoramic image. In this work, we introduce PaGeR (Panoramic Geometry Reconstruction), a framework to lift powerful 3D foundation models designed for perspective imagery to the panorama domain. Our strategy is to start from a pre-trained transformer for 3D reconstruction and turn it into a unified high-performance model that predicts scale-invariant depth, metric depth, surface normals, and sky masks from both perspective and omnidirectional images, in a single forward pass. By keeping architectural changes to a minimum and mixing perspective and panoramic images during training, PaGeR retains the rich 3D prior of the underlying foundation model while learning

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

to also estimate geometrically consistent 360° scenes from single panoramas. We extensively test our method in both indoor and outdoor environments and find that it delivers state-of-the-art performance and excellent zero-shot performance across a wide range of scenes.

1 Introduction

Sensing and understanding the 3D structure of the surrounding world is important in many applications, ranging from virtual and augmented reality to autonomous driving and robotics. Scene depth and surface normals are two central geometric properties in that context: together, they describe the position and the local surface orientation at any point of the scene, providing a complete representation that supports graphics tasks like rendering and relighting as well as high-level perception tasks like spatial reasoning and path planning.

A particularly attractive, but also heavily ill-posed variant is to recover depth or surface normals from a single RGB image [12, 20], obviating the need for multi-view capture and camera pose estimation. Early attempts relied on limited datasets and convolutional backbones [5, 9], but large-scale data collection [37] and advances in neural architectures, most notably vision transformers and denoising diffusion models [53, 20], have greatly advanced monocular geometry estimation. Most recently, this trend has converged with learning-based multi-view reconstruction, leading to foundational feed-forward models [45, 26] capable of zero-shot, dense 3D reconstruction. From their massive training datasets, captured under diverse imaging conditions, these models acquire not only an understanding of multi-view geometry but also an elaborate prior of the world's 3D surface structure, which supports detailed, dense depth estimation from single views.

Yet, these models are designed for perspective images such that every image covers only a limited field of view, and many viewpoints must be aggregated and fused to build up spatial context and perceive the complete environment. Panoramic images, by construction, provide a full 360° view around the camera location, offering rich global context for holistic 3D understanding. However, high-quality panoramic datasets with metrically accurate depth and surface normals, needed to train panoramic reconstruction models, are laborious to collect and remain scarce. As a result, existing models tend to overfit to comparatively small datasets and struggle to generalize to unseen scenes. Another limitation is that existing models commonly represent panoramic images in equirectangular projection, which introduces serious geometric distortions. On the one hand, this means an extremely uneven sampling of the ray space (and, after unwarping, of the 3D environment). On the other hand, and perhaps more importantly, it means that one cannot easily employ transfer learning from models trained with perspective images.

We take a different route and explore cubemaps as our panorama representation. Rather than designing custom architectures applicable specifically to panoramas, we use this parametrization to repurpose state-of-the-art perspective foundation models for the panoramic domain. The cubemap representation has been used to adapt pre-trained, diffusion-based image generators to 360° imagery [18]. For our purposes, we prefer to build on top of deterministic, feed-forward foundation models. Besides avoiding the computational overhead of diffusion and the practical limitations of operating in a compressed latent space, models like DA3 [26] are already designed for (perspective) multi-view input. They offer a natural synergy with the cubemap format, as their geometric prior includes the integration of multiple viewing directions and is fundamentally stronger than previous, purely monocular schemes. To properly anchor the spatial context, we explicitly condition the architecture on camera parameters and introduce targeted modifications of the decoder to ensure distortion-free and seamless reconstruction across face boundaries. Furthermore, we propose to use a mixed training regime with both synthetic panoramas and real perspective imagery. This strategy allows the network to adapt to the 360° setting while remaining firmly grounded in real-world image statistics, thus preventing overfitting to peculiarities of synthetic data and preserving the prior of the pre-trained foundation model.

Taken together, we introduce *PaGeR*, a unified geometry estimation framework for panoramic (and perspective) images. Its latent representation, inherited from the geometry foundation model, is holistic and allows for simultaneous decoding of multiple scene properties. We exploit this property and equip the backbone with multiple, coupled task heads: Scale-Invariant (SI) depth, metric scale estimation, surface normals, and sky segmentation. That design allows the network to jointly reason

about related geometric properties and to efficiently extract a comprehensive 3D representation in a single forward pass (see Fig. 1). By reparameterizing panoramic geometry as a structured multi-view problem, we achieve high-resolution, metrically accurate predictions that set a new state of the art for several benchmarks. Furthermore, to address the lack of benchmark data for rigorous evaluation in long-range, outdoor scenarios, we curate and introduce ZüriPano, the first real-world outdoor panoramic dataset, acquired through high-accuracy and high-resolution LiDAR scans in an urban environment. In summary, our contributions are:

- **A novel strategy to adapt geometry foundation models** to panorama geometry. Our scheme is built around the cubemap representation consisting of six perspective images, and combines it with a hybrid training strategy to seamlessly transfer 3D scene priors to the 360° panorama setting, sidestepping degradations caused by equirectangular distortion.
- **PaGeR, a unified panoramic geometry estimation model**, featuring a shared transformer backbone (adopted from DA3) and specialized task heads to enable holistic reconstruction in a single forward pass.
- **Zero-shot generalization** to unseen indoor and outdoor scenes, outperforming methods limited to a specific domain; and the new **ZüriPano** benchmark for zero-shot evaluation in urban setting.

2 Method

This section outlines the geometric preliminaries of panoramic representations and our perspective backbone architecture. We then introduce the panoramic adaptation layers and the hybrid training strategy designed to bridge the perspective and spherical domains. Finally, we detail the unified multi-task architecture and formalize the loss objectives for each geometric modality.

2.1 Preliminaries

Panoramic Image Representations. Panoramas capture a holistic $360^\circ \times 180^\circ$ environment on the unit sphere $\mathbb{S}^2$. The standard Equirectangular Projection (ERP) maps spherical coordinates, i.e., longitude $\theta \in [-\pi, \pi]$ and latitude $\phi \in [-\pi/2, \pi/2]$, to a 2D planar grid $(u, v) \in [0, 1]^2$ via:

$$u = \frac{\theta}{2\pi} + 0.5, \quad v = \frac{\phi}{\pi} + 0.5 \tag{1}$$

While structurally simple, ERP introduces severe nonlinear distortions. The horizontal sampling density scales by $\sec(\phi)$, causing extreme stretching near the poles ($\phi \to \pm\pi/2$). This domain shift degrades the efficacy of translation-invariant architectures optimized for perspective imagery.

To mitigate polar distortions, the cubemap projection maps $\mathbb{S}^2$ onto the six faces of a circumscribed unit cube $\mathcal{C}$. Each cube face constitutes a standard 90° FoV perspective image. A 3D ray $\mathbf{p} = (x, y, z) \in \mathbb{S}^2$ is mapped to local face coordinates via gnomonic projection (e.g., $u_c = x/z, v_c = y/z$ for the front face where $z = 1$). This piecewise perspective formulation offers uniform sampling and directly aligns with the inductive priors of models trained on perspective data. However, partitioning the continuous sphere introduces geometric and photometric discontinuities at face boundaries, requiring custom adaptations of the architecture to maintain global consistency.

Geometry Transformer Backbone. Our framework is compatible with any multi-view transformer architecture. We instantiate our model using Depth Anything 3 (DA3) [26], which couples a vision transformer encoder [30] with a dense prediction transformer decoder [38]. Given a set of S perspective views $\mathcal{I} = \{I_i\}_{i=1}^S$, the encoder tokenizes the inputs and routes them through interleaved intra-image and cross-image attention layers. This global attention mechanism can optionally be conditioned on explicit camera parameters, namely intrinsic matrices $\mathbf{K}_i \in \mathbb{R}^{3\times3}$ and extrinsic poses $\mathbf{E}_i \in SE(3)$, to guide spatial cross-view reasoning. The encoder yields hierarchical feature maps $\mathcal{F} = \{F^{(\ell)}\}$ across transformer layers ℓ, which the decoder progressively upsamples and fuses to output dense spatial predictions.

2.2 Panoramic Adaptation and Joint Training

To adapt the multi-view architecture for holistic 360° estimation, we format the panoramic input as a six-face cubemap and supply fixed camera matrices $\mathbf{K}$ alongside axis-aligned extrinsics $\mathbf{E}_i$,

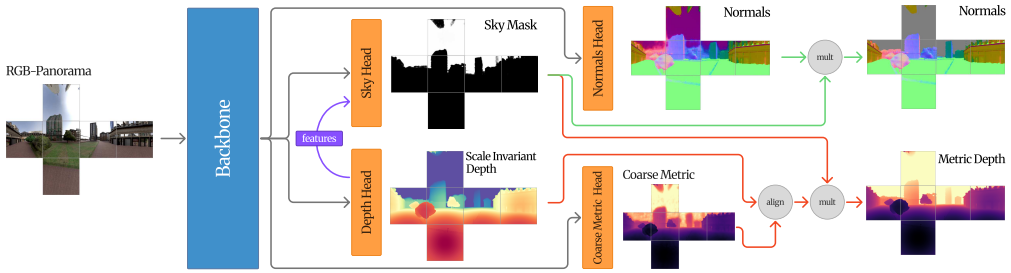

Figure 2: **PaGeR Architecture.** An input RGB panorama in cubemap format is processed by a shared geometry transformer backbone to predict a sky mask, scale-invariant (SI) depth, surface normals, and coarse metric depth. In the metric branch, the final absolute depth is obtained by aligning the SI depth with coarse metric predictions and masking the sky. In the normal branch, predicted orientations are masked by the sky segmentation to produce the final surface normal map.

$i = 1, \ldots, 6$. While these geometric parameters explicitly define the spatial configuration, naively assembling independent face predictions into an equirectangular projection yields pronounced discontinuities at the boundaries. Furthermore, training exclusively on synthetic panoramas can cause the model to quickly diverge from its pre-training weights. We resolve these challenges through structural adaptations that favor global feature extraction and local decoding, complemented by a regularized joint training regime.

Implicit Encoder Synchronization. We fine-tune the ViT encoder on panoramic data without any structural modifications. Guided by the fixed camera tokens, face positional embeddings, and cross-view attention layers, the network naturally learns to route context and synchronize features across adjacent cubemap faces. The fine-tuning allows the backbone to adapt to the spherical topology while preserving the rich perspective priors learned during pre-training.

Spherically Aware Decoder Padding. Although global synchronization occurs in the encoder, local boundary artifacts can still emerge during dense upsampling in the decoder. To ensure continuous spherical sampling, we integrate cross-face valid padding into all convolutional and interpolation operations within the decoder architecture [15]. Instead of standard zero padding, this layer dynamically extracts features from geometrically adjacent cubemap faces, enforcing seamless geometric and photometric transitions across all boundaries.

Mixed Panoramic / Perspective Co-Training. To preserve the rich priors inherited from the perspective backbone and mitigate the sim-to-real domain gap, we employ a training strategy that alternates between two data streams. For panoramic batches, the network processes the full six-face configuration ($S = 6$) with active cross-face padding. For perspective batches, we isolate a single real-world image ($S = 1$), warp it to a 90° field of view to match the imaging geometry of cubemap faces, and assign it the extrinsics of a random equatorial face. Cross-face padding is dynamically disabled for these perspective samples, with the layer reverting to standard zero padding. The dual-stream training protects the model from catastrophic forgetting while teaching it to handle continuous spherical observations.

2.3 Multi-Task Geometric Decoding

Existing geometric foundation models are typically confined to a single modality, such as scale-invariant depth estimation. For a comprehensive 3D understanding of 360° environments, we add multi-task decoding to the unified backbone. Specialized prediction heads simultaneously decode depth, surface orientation, and sky masks in a single forward pass (see Fig. 2), always operating on the planar cubemap faces to benefit from the underlying perspective prior.

Scale-Invariant Depth. The model is supervised with the local, orthogonal per-face log-planar depth $z^* = \log(Z^*_{\text{planar}})$ to compress metric variance and avoid optimization bias from distant background objects. We remove the final exponential activation of the decoder to work directly in its native log space. The head outputs both the predicted scale-invariant log depth $\hat{z}_{\text{SI}}$ and an aleatoric confidence map c_p. To isolate relative shape from metric size, we dynamically compute an optimal log-space shift $\beta^* = \arg\min_\beta \|(\hat{z}_{\text{SI}} + \beta) - z^*\|_2^2$ and optimize the aligned predictions $\hat{z}_{\text{aligned}} = \hat{z}_{\text{SI}} + \beta^*$. We supervise the scale-invariant depth branch using a composite loss function $\mathcal{L}_{\text{depth}}$ that balances

per-pixel precision, local smoothness, and surface alignment:

$$\mathcal{L}_{\text{depth}} = \lambda_{L_1}\mathcal{L}_1 + \lambda_{\text{grad}}\mathcal{L}_{\text{grad}} + \lambda_{\text{norm}}\mathcal{L}_{\text{norm}} \tag{2}$$

where $\mathcal{L}_1 = \frac{1}{N}\sum_p \left(c_p \left|\hat{z}_{\text{aligned},p} - z_p^*\right| - \lambda_c \log c_p\right)$, $\mathcal{L}_{\text{grad}} = \frac{1}{N}\sum_p \sum_{i\in\{x,y\}} \left|\partial_i \hat{z}_{\text{aligned},p} - \partial_i z_p^*\right|$ and $\mathcal{L}_{\text{norm}} = \frac{1}{N}\sum_p \left(1 - \hat{\mathbf{n}}_p \cdot \mathbf{n}_p^*\right)$. Here, N denotes the total number of valid pixels. The primary loss $\mathcal{L}_1$ measures the absolute discrepancy scaled by the predicted aleatoric confidence c_p. We complement this with an edge-aware gradient penalty $\mathcal{L}_{\text{grad}}$ to preserve discontinuities at object boundaries, and a normal consistency loss $\mathcal{L}_{\text{norm}}$ that enforces geometric alignment via the cosine similarity between ground-truth orientations $\mathbf{n}^*$ and surface normals $\hat{\mathbf{n}}$, derived analytically from the predicted depth maps.

Surface Normals. We instantiate a dedicated, parallel decoding branch for normals. It is initialized with the pre-trained depth weights to benefit from the close connection between depth and normals. The final layer is modified to output three-dimensional unit vectors $\hat{\mathbf{n}}$. Training utilizes a joint objective $\mathcal{L}_{\text{normal}} = \lambda_{\text{cos}}\mathcal{L}_{\text{cos}} + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}}$, which combines a pixel-wise cosine similarity loss with a VGG-based perceptual loss [41]. The latter serves to prevent over-smoothing and promote sharp edges.

Metric Scale. To reconstruct an absolute scale without disrupting the reconstruction of relative local geometry, we decouple metric estimation from the high-resolution, scale-invariant branch. A parallel, coarse decoder predicts a low-resolution metric log-depth map $\hat{z}_{\text{m}}$ alongside an aleatoric confidence map $\hat{c}_{\text{m}}$. From that map, we infer a global scale factor $\hat{\beta}$ as the median difference between the coarse metric log-depth and an average-pooled version of its scale-invariant counterpart, computed over a lower-resolution grid of spatial anchors $\mathbf{a}$:

$$\hat{\beta} = \text{median}_{\mathbf{a}}\Big(\hat{z}_{\text{m}}(\mathbf{a}) - \text{pool}[\hat{z}_{\text{SI}}](\mathbf{a})\Big) \tag{3}$$

The median filters out localized geometric discrepancies. The final, absolute metric depth is recovered as $\hat{Z}_{\text{m}} = \exp(\hat{\beta})\hat{Z}_{\text{SI}}$. The metric head is trained with a coverage-weighted version of the confidence-aware $\mathcal{L}_1$ loss against appropriately downsampled ground-truth targets, ensuring that invalid regions do not corrupt the scale estimation.

Sky Segmentation. Modeling infinite depth directly destabilizes metric regression. We explicitly decouple unbounded regions by introducing a lightweight sky segmentation branch, such that the primary depth heads can focus on structures with finite depth. The branch reads out geometric cues from intermediate decoder features and fuses them with semantic tokens extracted from the deep encoder layers and passed through a small, fully connected network. The concatenated feature maps are mapped to binary sky probabilities $\hat{Y}$ with a shallow convolutional decoder. This head is trained with a combination of binary cross-entropy, focal [27], and dice losses [43] w.r.t. the ground-truth mask. Its outputs serve to mask sky regions with undefined geometry in the depth and normal outputs.

3 Experiments

We evaluate PaGeR across diverse quantitative and qualitative experiments on both indoor and outdoor environments. We compare against existing state-of-the-art panoramic geometry estimators and provide detailed ablation studies to isolate and validate the individual structural adaptations and joint training choices of our approach.

3.1 Training Details

We initialize our framework from pre-trained DA3 weights [26] featuring a DINOv2 backbone [30]. Optimization proceeds in two sequential stages. First, we jointly train the scale-invariant depth and surface normal decoders, adapting the backbone features to support both geometric modalities. Second, we freeze these components and independently train the metric scale and sky segmentation heads using the frozen feature representations. We optimize using AdamW [29] with an exponentially decaying learning rate schedule initialized at $3 \cdot 10^{-4}$ and an Exponential Moving Average decay of 0.999. The first stage requires 12 hours of training on 8 NVIDIA H200 GPUs, while the second stage completes in an additional 8 hours.

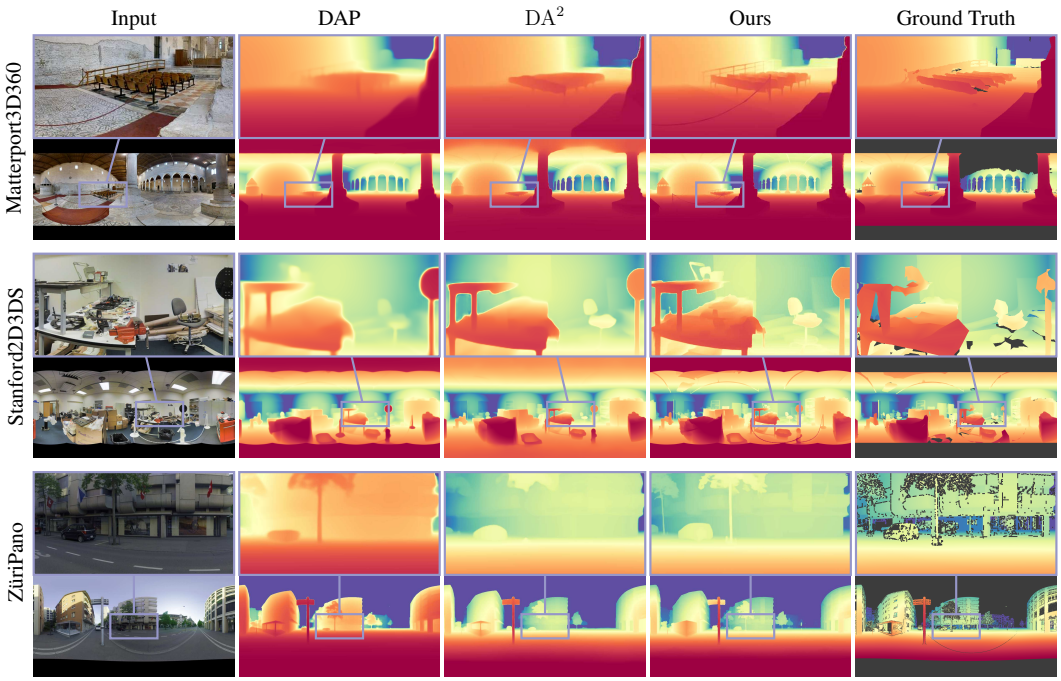

Figure 3: **Qualitative comparison of panoramic depth estimation.** Visual results from PaGeR, DAP [28], and DA^2 [24] (the strongest metric and scale-invariant baselines) alongside the RGB input and ground-truth depth on Matterport3D360, Stanford2D3DS, and ZüriPano. Our framework recovers sharper boundaries and more accurate global structures than competitors. Additional examples are in the appendix. Best viewed zoomed in.

	Matterport3D360			**Stanford2D3DS**			**ZüriPano**			**Holo360D**		
Method	AbsRel ↓	RMSE ↓	δ_1 ↑	AbsRel ↓	RMSE ↓	δ_1 ↑	AbsRel ↓	RMSE ↓	δ_1 ↑	AbsRel ↓	RMSE ↓	δ_1 ↑
DreamCube* [15]	30.45	108.81	54.26	28.45	75.30	56.39	29.57	484.91	53.07	–	–	–
DepthAnyCamera [10]	25.62	94.06	68.11	18.35	53.73	76.23	21.98	487.66	65.41	–	–	–
MoGe [47]	18.12	81.68	77.11	18.12	81.68	77.11	19.25	484.18	77.12	–	–	–
EGformer* [57]	16.74	96.15	79.32	13.64	58.95	84.64	55.49	721.83	31.21	–	–	–
DAP [28]	15.84	85.18	82.44	9.76	46.69	92.37	19.86	583.42	72.09	42.70	8.44	34.30
RPG360 [17]	15.40	79.40	82.40	11.91	46.60	87.73	<u>18.27</u>	<u>455.28</u>	<u>78.41</u>	–	–	–
UniK3D† [33]	14.82	69.27	83.67	9.93	40.69	93.34	31.00	832.13	38.78	53.80	8.70	23.70
PanDA* [4]	13.84	82.82	84.26	10.69	55.72	90.73	24.64	515.48	58.31	<u>39.90</u>	<u>6.16</u>	<u>45.90</u>
DA^2 [24]	<u>11.06</u>	<u>67.72</u>	<u>89.18</u>	<u>7.64</u>	<u>37.29</u>	<u>95.63</u>	61.22	869.08	2.17	51.60	7.34	11.10
PaGeR	**9.67**	**64.69**	**90.87**	**5.93**	**35.34**	**96.10**	**9.36**	**299.61**	**94.75**	**18.70**	**3.54**	**83.00**

Table 1: **Quantitative comparison between PaGeR and state-of-the-art baselines across indoor (Matterport3D360, Stanford2D3DS) and outdoor (ZüriPano, Holo360D)**. Best and second-best results are indicated in **bold** and <u>underlined</u>, respectively. Methods optimized using in-domain training are marked with †, and affine-invariant methods are denoted with *.

Our mixed data regime balances 80k synthetic panoramas from Structured3D [59] and our PanoInfinigen dataset with 10k real perspective images from ScanNet++ [54] and ARKitScenes [2] to mitigate the sim-to-real domain gap. Following standard practice [28, 51], we train independent metric scale heads for indoor and outdoor environments to accommodate distinct spatial layouts. We maintain a training resolution of 504×504 pixels per cubemap face, which assembles into a 2K equirectangular panorama. At inference time, our unified framework processes a full 2K panorama in 0.5 seconds while consuming 12.8 GB of memory, allowing for deployment on a single consumer-grade GPU. The choices of hyperparameters are given in Tab. 8.

3.2 Evaluation Protocol

Datasets and Evaluation Ranges. Consistent with panoramic benchmarks [4, 46, 24], we evaluate scale-invariant and metric depth on the real-world indoor datasets Matterport3D360 [39] and

Method	Matterport3D360			Stanford2D3DS			ZüriPano			Holo360D		
	AbsRel ↓	RMSE ↓	δ_1 ↑	AbsRel ↓	RMSE ↓	δ_1 ↑	AbsRel ↓	RMSE ↓	δ_1 ↑	AbsRel ↓	RMSE ↓	δ_1 ↑
UniK3D† [33]	33.43	143.24	46.48	24.37	64.71	64.50	35.81	1345.48	34.86	51.30	9.37	28.50
RPG360 [17]	29.26	136.99	35.33	29.35	90.73	17.36	40.36	772.16	2.14	–	–	–
DepthAnyCamera [10]	25.16	132.66	62.25	17.15	59.29	77.98	33.23	716.38	37.83	27.40	6.79	61.20
DAP‡ [28]	30.16	168.88	**80.19**	10.97	53.39	90.64	47.00	839.72	24.02	59.30	11.05	56.90
PaGeR‡	**21.83**	**123.48**	69.50	**10.94**	**45.43**	**90.94**	**31.97**	**530.85**	**39.30**	**23.50**	**4.04**	**72.00**

Table 2: **Quantitative comparison of panoramic metric depth models**. Methods optimized using in-domain training are marked with †, while those using separate indoor or outdoor prediction heads are marked with ‡. **Bold** font marks the best result, underlined the second-best.

Stanford2D3DS [42]. To address the indoor bias of existing literature, we also introduce ZüriPano, a custom outdoor urban LiDAR dataset tailored for long-range geometric evaluation (see Sec. B). Additionally, we evaluate on Holo360D [31], a synthetic outdoor dataset. For all depth evaluations, we enforce a broad range constraint of $[0, 75]$ m to ensure a thorough assessment of global structures. This avoids the evaluation bias of prior works [4, 24] that use a narrow $[0, 5]$ m window, masking far-field errors. For completeness, we report the results for the near-field depth evaluation in the supplementary J. For surface orientation, we benchmark on the Structured3D dataset [59], where all compared baselines are trained to guarantee a fair comparison.

Metrics and Processing. Following established conventions [53], depth accuracy is measured via Absolute Relative Error (AbsRel), Root Mean Squared Error (RMSE), and the threshold percentage δ_1. Scale-invariant depth maps are adjusted using a standard least-squares alignment prior to scoring, whereas metric depth is evaluated directly without modifications. For surface normals, we report the Mean Angular Error, Mean Squared Error (MSE), and the fraction of pixels with errors below $\delta_\theta \in \{5°, 22.5°\}$ [13], with all predictions normalized to unit length before evaluation. All reported numbers for competing methods have been recalculated to evaluate at the native resolution of each dataset; furthermore, to ensure fairness to low-resolution methods, we additionally evaluate in a low-resolution setting of 512×1024 by downsampling the ground truth, with these additional results provided in the supplementary J. The same ground-truth sky masks are used throughout the whole evaluation, for each evaluated method.

3.3 Quantitative Comparison

As demonstrated in Table 1, PaGeR consistently outperforms existing methods across all datasets. While it demonstrates notable improvements on indoor-biased benchmarks, its primary advantage lies in its cross-domain generalization. On the challenging outdoor ZüriPano dataset, PaGeR reduces the Absolute Relative Error (AbsRel) from the previous best 18.27 for RPG360 to 9.36, nearly cutting it in half. This substantial improvement confirms that our framework enhances structural geometry globally and does not need to trade off indoor vs. outdoor accuracy.

This balanced capability extends directly to absolute scale recovery, as detailed in Table 2. By decoupling metric scale estimation from the structural backbone, our independent domain heads successfully specialize to indoor or outdoor structures while sharing the same underlying transformer features. On the ZüriPano metric benchmark, PaGeR establishes a commanding lead with an RMSE of 530.85 compared to 716.38 for the next best, DepthAnyCamera. At the same time, it maintains high accuracy indoors, outperforming recent baselines such as UniK3D and DAP on both indoor datasets.

To evaluate higher-order geometric consistency, we report surface normal estimation on the Structured3D dataset in Table 3. PaGeR sets a new state of the art, outperforming specialized architectures including PanoNormal and HyperSphere. Specifically, our framework achieves a Mean Angular Error of $5.49°$ and an MSE of 174.9, which represents a major reduction from the 246.6 MSE of the previous state of the art. This validates our multi-task architecture choice, demonstrating that joint learning across separate heads improves the recovery of fine-grained surface structures.

3.4 Qualitative Comparison

Visual comparisons across both scale-invariant and metric depth settings are presented in Figure 3. While baselines such as DAP and DA^2 suffer from structural over-smoothing and fail to resolve fine details, PaGeR delivers sharp geometric boundaries alongside a globally continuous scene layout.

Method	Replica360_4K (Seam Eval.) SDD ↓	SP ↓	SS ↓	Stanford2D3DS (SI Depth Eval.) AbsRel ↓	RMSE ↓	δ_1 ↑
DA3	33.71	44.29	68.97	6.58	35.79	95.94
PaGeR	**0.94**	**0.38**	**2.31**	**5.93**	**35.34**	**96.10**

(a) Comparison against DA3 baseline.

Method	Mean ↓	MSE ↓	δ_{5° ↑	$\delta_{22.5^\circ}$ ↑
Frozen backbone	6.21	251.1	77.00	90.52
W/o perceptual	5.92	198.2	77.82	90.96
Rand. init. DPT	6.02	191.9	78.02	89.76
Full PaGeR	**5.49**	**174.9**	**79.91**	**92.83**

(b) Surface Normals on Structured3D.

Table 4: **Ablations for vanilla DA3 and surface normals**.

This structural precision is further highlighted in the point cloud reconstructions illustrated in Figure 9. Here, competing approaches frequently introduce warped geometries and surface discontinuities along the boundaries. In contrast, our framework accurately recovers complex outdoor topographies while preserving intricate indoor structural arrangements. Furthermore, Figure 4 demonstrates the high-resolution detail of our estimated surface normals; for more qualitative examples, see Appendix K.

3.5 Ablation Studies

Comparison to Baseline. We evaluate PaGeR against the vanilla DA3 baseline using depth accuracy and cross-face geometric consistency metrics (Table 4a; formal definitions in Section D). PaGeR significantly improves upon both. While preserving the robust geometric priors of the base model, our framework resolves depth misalignment across face boundaries to yield seamless reconstructions, as qualitatively illustrated in Figure 8. Apart from the standard 6-face cubemap input, we ablate alternative input configurations to the DA3 backbone, provided in the supplementary material G.

Method	AbsRel ↓	RMSE ↓	δ_1 ↑
W/o camera cond.	10.12	43.91	90.01
Linear-depth	6.96	39.12	94.98
W/o sky segm. head	6.88	40.74	93.59
Euclidean-depth	6.34	36.91	95.48
W/o valid padding	6.45	37.00	95.73
W/o structural losses	6.11	36.72	95.93
Vanilla L1 loss	6.15	36.02	96.00
W/o perspective data	6.45	37.98	95.45
PaGeR	**5.93**	**35.34**	**96.10**

(a) Scale-invariant depth.

Method	AbsRel ↓	RMSE ↓	δ_1 ↑
Single Global Scale	12.67	47.14	88.05
Dense, F=1	11.35	46.31	88.35
Dense, F=2	11.34	45.98	90.01
Dense, F=4 (Ours)	**10.94**	**45.43**	**90.94**
Dense, F=8	12.15	47.09	89.92

(b) Metric depth.

Table 5: **Ablations for scale-invariant and metric depth on Stanford2D3DS**.

Scale-Invariant Depth. Table 5a isolates the impact of individual architectural choices on the Stanford2D3DS dataset. Omitting explicit camera conditioning causes the most severe performance drop, increasing absolute relative error. Our log-space formulation also outperforms alternative linear-depth targets. Furthermore, decoupling infinite depth is critical; removing the auxiliary sky segmentation head causes a major spike in RMSE. Finally, simplifying the training objective or omitting structural data refinements, such as cross-face valid padding and joint perspective training, consistently degrades overall precision.

Method	Mean ↓	MSE ↓	δ_{5° ↑	$\delta_{22.5^\circ}$ ↑
UniFuse [16]	8.25	453.1	76.24	87.55
PanoFormer [40]	16.92	1053.5	59.13	75.50
OmniFusion [25]	20.70	832.0	28.51	63.55
MonoViT [58]	5.92	277.8	78.93	90.58
HyperSphere [19]	5.79	253.4	78.38	90.73
MTL [14]	8.98	469.0	72.51	86.02
PanoNormal [40]	5.56	246.6	79.18	91.01
PaGeR	**5.49**	**174.9**	**79.91**	**92.83**

Table 3: **Quantitative comparison of surface normals on the Structured3D dataset**. All evaluated methods are in-domain architectures optimized directly on Structured3D.

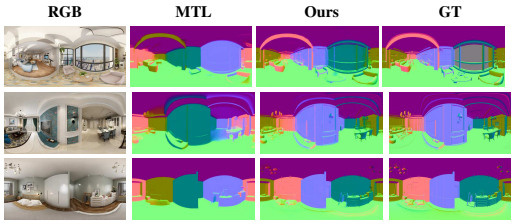

Figure 4: **Qualitative comparison of panoramic surface normals estimation.** Results from PaGeR and MTL (strongest baseline), shown alongside the RGB input and ground-truth depth on panoramas from the Structured3D dataset. (Best viewed zoomed in.)

Surface Normals. We validate our surface normal setup on the Structured3D dataset (Table 4b). Altering the encoder configuration proves highly detrimental; freezing the backbone causes a major spike in MSE, while optimizing a randomly initialized decoder yields the poorest threshold accuracy. Omitting the perceptual loss from the training objective causes a clear regression across all metrics, confirming that feature matching regularizers are necessary to prevent over-smoothing and recover sharp geometric boundaries.

Metric Scale. Table 5b evaluates our metric supervision strategy on the Stanford2D3DS dataset. Relying strictly on global scalar estimation rather than dense supervision across distributed anchor points yields the poorest baseline performance. Within our dense supervision framework, tuning the feature downsampling factor F is critical; both insufficient ($F = 1$) and extreme ($F = 8$) downsampling undermine accuracy. Our final configuration successfully balances these factors to achieve optimal metric depth reconstruction.

Architectural vs. Training Adaptations

Our architectural adaptation (Spherically-Aware Decoder Padding, SADP) and mixed panoramic co-training target distinct failure modes. SADP is purely local: it replaces zero padding with features from geometrically adjacent faces, ensuring they agree exactly *at* the boundary, but cannot force independent predictions to cohere as a unified scene. Conversely, panoramic co-training addresses global consistency, teaching the network to align the six faces into a single spatial layout with a unified metric scale.

Table 6 isolates these contributions. SADP alone nearly halves seam defects (SDD) but barely improves depth accuracy (AbsRel). Panoramic co-training alone improves accuracy slightly, but seam artifacts persist. Only their combination yields optimal results. Notably, the absolute error improvements appear small because the unadapted perspective foundation model already provides a highly efficient baseline (6.58 AbsRel)—explaining why our approach achieves state-of-the-art performance with only $\sim$100k panoramas, while native equirectangular training requires orders of magnitude more data.

Variant	**AbsRel** $\downarrow$ (Stanford2D3DS)	**Seam Defects (SDD)** $\downarrow$ (Replica360-4K)
Vanilla DA3 on cubemaps (No adaptation)	6.58	33.71
SADP only, no panoramic training	6.49	19.01
Panoramic training only, without SADP	6.45	12.67
Panoramic training + SADP (Ours)	**5.93**	**0.94**

Table 6: **Ablation of local structural adaptation (SADP) versus global panoramic co-training.** Metrics are scaled by $\times 100$. Both adaptations are necessary to achieve optimal global accuracy and eliminate local seam artifacts.

4 Related Work

Perspective Geometry Estimation. Monocular depth estimation has evolved into general-purpose architectures capable of robust zero-shot 3D geometry inference [37, 34, 53], recently augmented by diffusion priors [20, 7, 8] and unified geometric models [12, 47] for joint depth and normal prediction. Despite high structural fidelity, recovering absolute metric scale remains challenging. Naively regressing dense metric depth or localized scales [3, 50, 55] disrupts perspective priors, while directly regressing a single global scale factor [60] suffers from sparse gradients. However, for panoramic imagery with fixed 360° intrinsics, metric reconstruction mathematically reduces to estimating a single global scale. Exploiting this, we propose a framework that transfers rich perspective priors to panoramas, outputting scale-invariant geometry alongside a robustly decoupled global metric scale.

Visual Geometry Foundation Models. Visual Geometry Foundation Models (VGFMs) [48, 23, 45, 26, 6] have shifted 3D reconstruction from task-specific estimators to unified, feed-forward architectures. By treating reconstruction as dense correspondence regression, they bypass traditional iterative Structure-from-Motion. A key advantage of VGFMs is their input flexibility, seamlessly transitioning from monocular to multi-view regimes and natively incorporating varying camera

intrinsics or extrinsics. We build directly upon these robust, flexible priors, adapting their multi-view feed-forward capabilities to the panoramic domain.

Panoramic Geometry Estimation. Panoramic geometry estimation historically struggles with equirectangular distortions and data scarcity. Previous methods range from designing specialized architectures [44, 40, 13, 21] to adapting perspective models via multi-projection formats [46, 1]. Recently, training-free cubemap adaptations [18, 15, 17] mitigate distortions by predicting mutually consistent faces, though they often struggle to merge views back into a seamless equirectangular projection (ERP). Consequently, the field faces a trade-off: training-free multi-face methods exhibit seam artifacts, while fully-supervised continuous ERP models [24, 28] incur prohibitive training costs. Even concurrent efforts adapting VGFMs to 360° panoramas [11, 56] require extensive retraining. In contrast, our approach retains and reuses the original perspective VGFM priors, extending them to panoramas without requiring massive retraining or sacrificing expressivity. We additionally provide extensive analysis and the comparison of our method to similar and relevant prior works in the supplementary C.

5 Conclusion

We have presented PaGeR, a unified framework for panoramic geometry estimation that successfully lifts the robust representations of perspective depth models into the spherical domain. By combining a synchronized cubemap representation with separate decoding heads, our architecture transfers established perspective priors to multiple 360° geometric tasks without requiring extensive retraining. To guide this adaptation, we introduced a joint panoramic/perspective training regime. Extensive evaluations demonstrate that PaGeR establishes a new state of the art in zero-shot panoramic reconstruction across diverse indoor and outdoor environments. Although our instantiation leverages DA3, the architectural principles of our framework maintain compatibility with alternative geometric transformers [45, 49], offering a generalized methodology for bridging the domain gap between perspective and panoramic 3D scene understanding.

6 Limitations

While the unbounded 360° context of panoramas reduces global structural ambiguities—such as truncated objects or unconstrained floor layouts—it cannot resolve local pixel ambiguities inherent to monocular depth prediction. First, PaGeR struggles with specular, reflective, or transparent surfaces, typically reconstructing the geometry reflected or seen through them (e.g., interpreting a mirror as an adjoining room). Resolving this likely requires a fundamentally different algorithmic approach. Second, accuracy naturally degrades in large textureless regions or under out-of-distribution conditions (e.g., complex outdoor lighting), where the model must rely on generalized layout priors. Furthermore, our outdoor evaluation is constrained by the scarcity of comprehensive, real-world panoramic benchmarks, relying primarily on a limited set of 100 European urban panoramas (ZüriPano) alongside synthetic data (Holo360D). Consequently, we are currently unable to fully validate the model's zero-shot generalization across highly diverse, natural, or non-urban outdoor landscapes, representing a shared limitation in both our evaluation and the broader field.

Finally, despite cross-face padding and global attention mechanisms, subtle geometric misalignments can occasionally emerge at face boundaries in structurally complex scenes. This is primarily a data limitation, as the network's attention must cover a long tail of complex edge configurations. While failures in textureless regions, out-of-distribution appearances, and seam artifacts are expected to improve with dataset scaling, guaranteeing absolute boundary consistency may ultimately require explicit mathematical continuity constraints in future work.

Acknowledgments

This work was supported by ETHAR (funded by a Google donation) and the Swiss National Supercomputing Center (Swiss AI Initiative, project a144). We thank Hothifa Smair and Parametra for authorizing the use of the iCity Blender add-on to generate and release the *PanoInfinigen* urban split under an academic license. We also thank Veljko Bozic for assembling the urban scenes.

References

[1] Hao Ai and Lin Wang. Elite360D: Towards efficient 360 depth estimation via semantic-and distance-aware bi-projection fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, and Elad Shulman. ARKitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile RGB-D data. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2021.

[3] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *preprint arXiv:2302.12288*, 2023.

[4] Zidong Cao, Jinjing Zhu, Weiming Zhang, Hao Ai, Haotian Bai, Hengshuang Zhao, and Lin Wang. PanDA: Towards panoramic depth anything with unlabeled panoramas and mobius spatial augmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[5] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[6] Xianze Fang, Jingnan Gao, Zhe Wang, Zhuo Chen, Xingyu Ren, Jiangjing Lyu, Qiaomu Ren, Zhonglei Yang, Xiaokang Yang, Yichao Yan, and Chengfei Lyu. Dens3r: A foundation model for 3d geometry prediction. In *The Fourteenth International Conference on Learning Representations*, 2026.

[7] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *European Conference on Computer Vision (ECCV)*, 2024.

[8] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan De Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[9] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[10] Yuliang Guo, Sparsh Garg, S Mahdi H Miangoleh, Xinyu Huang, and Liu Ren. Depth any camera: Zero-shot metric depth estimation from any camera. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[11] Yijing Guo, Mengjun Chao, Luo Wang, Tianyang Zhao, Haizhao Dai, Yingliang Zhang, Jingyi Yu, and Yujiao Shi. PanoVGGT: Feed-forward 3d reconstruction from panoramic imagery. *preprint arXiv:2603.17571*, 2026.

[12] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[13] Kun Huang, Fanglue Zhang, and Neil A Dodgson. PanoNormal: Monocular indoor 360° surface normal estimation. *preprint arXiv:2405.18745*, 2024.

[14] Kun Huang, Fanglue Zhang, Fangfang Zhang, Yu-Kun Lai, Paul L Rosin, and Neil A Dodgson. Multi-task geometric estimation of depth and surface normal from monocular 360° images. *preprint arXiv:2411.01749*, 2024.

[15] Yukun Huang, Yanning Zhou, Jianan Wang, Kaiyi Huang, and Xihui Liu. DreamCube: 3d panorama generation via multi-plane synchronization. *preprint arXiv:2506.17206*, 2025.

[16] Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang. Unifuse: Unidirectional fusion for 360° panorama depth estimation. *IEEE Robotics and Automation Letters*, 2021.

[17] Dongki Jung, Jaehoon Choi, Yonghan Lee, and Dinesh Manocha. RPG360: Robust 360 depth estimation with perspective foundation models and graph optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026.

[18] Nikolai Kalischek, Michael Oechsle, Fabian Manhardt, Philipp Henzler, Konrad Schindler, and Federico Tombari. CubeDiff: Repurposing diffusion-based image models for panorama generation. In *International Conference on Learning Representations (ICLR)*, 2025.

[19] Antonis Karakottas, Nikolaos Zioulis, Stamatis Samaras, Dimitrios Ataloglou, Vasileios Gkitsas, Dimitrios Zarpalas, and Petros Daras. 360° surface regression with a hyper-sphere loss. *preprint arXiv:1909.07043*, 2019.

[20] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[21] Jongsung Lee, Harin Park, Byeong-Uk Lee, and Kyungdon Joo. HUSH: Holistic panoramic 3d scene understanding using spherical harmonics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[22] Leica Geosystems. *Leica RTC360 3D Reality Capture Solution System Specification*. Hexagon AB, 2026. Accessed: 2026-05-02.

[23] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with MASt3R. In *European Conference on Computer Vision (ECCV)*, 2024.

[24] Haodong Li, Wangguangdong Zheng, Jing He, Yuhao Liu, Xin Lin, Xin Yang, Ying-Cong Chen, and Chunchao Guo. DA^2: Depth anything in any direction. In *International Conference on Learning Representations (ICLR)*, 2026.

[25] Yuyan Li, Yuliang Guo, Zhixin Yan, Xinyu Huang, Ye Duan, and Liu Ren. OmniFusion: 360 monocular depth estimation via geometry-aware fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[26] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations (ICLR)*, 2026.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[28] Xin Lin, Meixi Song, Dizhe Zhang, Wenxuan Lu, Haodong Li, Bo Du, Ming-Hsuan Yang, Truong Nguyen, and Lu Qi. Depth Any Panoramas: A foundation model for panoramic depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *preprint arXiv:1711.05101*, 2019.

[30] 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 Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 1, 2024.

[31] Jing Ou, Zidong Cao, Yinrui Ren, Zhuoxiao Li, Jinjing Zhu, Tongyan Hua, Shuai Zhang, Hui Xiong, and Wufan Zhao. Holo360d: A large-scale real-world dataset with continuous trajectories for advancing panoramic 3d reconstruction and beyond, 2026.

[32] Parametra. iCity, professional procedural city generation add-on for blender. `https://parametra.net/`, 2026. Accessed: 2026-05-02.

[33] Luigi Piccinelli, Christos Sakaridis, Mattia Segu, Yung-Hsu Yang, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniK3D: Universal camera monocular 3d estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[34] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *preprint arXiv:2502.20110*, 2025.

[35] Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[36] Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[37] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2020.

[38] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[39] Manuel Rey-Area, Mingze Yuan, and Christian Richardt. Matterport3D 360° RGBD dataset. `https://researchdata.bath.ac.uk/1126/`, 2022.

[40] Zhijie Shen, Chunyu Lin, Kang Liao, Lang Nie, Zishuo Zheng, and Yao Zhao. PanoFormer: panorama transformer for indoor 360° depth estimation. In *European Conference on Computer Vision (ECCV)*, 2022.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[42] Stanford Doerr School of Sustainability Data Repository. Stanford 2D-3D-Semantics dataset (2D-3D-S). `https://sdss.redivis.com/datasets/f304-a3vhsvcaf?v=1.0`, 2024.

[43] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. *Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations*, page 240–248. Springer International Publishing, 2017.

[44] Fu-En Wang, Yu-Hsuan Yeh, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. Bifuse++: Self-supervised and efficient bi-projection fusion for 360 depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5448–5460, 2022.

[45] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[46] Ning-Hsu Albert Wang and Yu-Lun Liu. Depth anywhere: Enhancing 360 monocular depth estimation via perspective distillation and unlabeled data augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[47] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[48] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[49] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning, 2026.

[50] Chengrui Wei, Meng Yang, Lei He, and Nanning Zheng. FS-Depth: Focal-and-scale depth estimation from a single image in unseen indoor scene. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(11), 2024.

[51] Tao Wen, Jiepeng Wang, Yabo Chen, Shugong Xu, Chi Zhang, and Xuelong Li. Metric-solver: Sliding anchored metric depth estimation from a single image, 2025.

[52] Ziyi Wu, Daniel Watson, Andrea Tagliasacchi, David J. Fleet, Marcus A. Brubaker, and Saurabh Saxena. 360anything: Geometry-free lifting of images and videos to 360 degrees, 2026.

[53] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[54] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3d indoor scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[55] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[56] Jiayi Yuan, Haobo Jiang, De Wen Soh, and Na Zhao. VGGT-360: Geometry-consistent zero-shot panoramic depth estimation. *preprint arXiv:2603.18943*, 2026.

[57] Ilwi Yun, Chanyong Shin, Hyunku Lee, Hyuk-Jae Lee, and Chae Eun Rhee. EGformer: Equirectangular geometry-biased transformer for 360 depth estimation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[58] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. MonoViT: Self-supervised monocular depth estimation with a vision transformer. In *2022 International Conference on 3D Vision (3DV)*, 2022.

[59] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A large photo-realistic dataset for structured 3d modeling. In *European Conference on Computer Vision (ECCV)*, 2020.

[60] Ruijie Zhu, Chuxin Wang, Ziyang Song, Li Liu, Jianfeng He, Jiacheng Deng, Tianzhu Zhang, and Yongdong Zhang. ScaleDepth: Decomposing metric depth estimation into semantic-aware scale prediction and adaptive relative depth estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, page 1–1, 2026.

A PanoInfinigen

High-quality datasets are the cornerstone of robust monocular depth estimation. While perspective depth estimation has benefited immensely from large-scale, diverse data collections, the panoramic domain suffers from limited training data. Existing standard benchmarks, such as Stanford2D3DS [42] and Matterport3D360 [39], rely on real-world scanners. Consequently, they are restricted to static indoor environments and often contain acquisition artifacts, such as missing depth values in reflective or distant regions. Synthetic alternatives like Structured3D [59] offer dense, complete ground truth but are similarly confined to low resolutions and indoor scenes with limited variability. To address this gap, we introduce *PanoInfinigen*, a large-scale synthetic dataset designed to unlock the potential of high-resolution, general-purpose panoramic depth estimation.

Data Generation and Characteristics. PanoInfinigen provides complete, high-quality ground truth for both depth and surface normals across a variety of indoor and outdoor scenes. The dataset is intended to support the development of high-performance models; moreover, it is extensible through our accompanying open-source generation tool. To construct PanoInfinigen, we build upon Infinigen [35, 36], a procedural content generation framework capable of synthesizing realistic, unbounded 3D environments. In contrast to methods that depend on fixed 3D asset libraries, Infinigen procedurally generates both geometry and textures, thereby enabling effectively unlimited variability in scene layout, object placement, and illumination conditions. We extend its rendering pipeline to support 360° equirectangular projection and generate 70,000 unique panoramas from 20,000 distinct scenes.

To extend coverage to urban environments, we employ the iCity [32] procedural city-generation add-on for Blender, which automates the creation of realistic and diverse synthetic urban scenes. We synthesize 20 urban environments with varying configurations and visual styles, and render approximately 400 panoramas per city using our existing Infinigen-based auto-rendering tool. This results in roughly 7,000 high-quality outdoor panoramas, each accompanied by dense depth and surface normal maps.

The resulting dataset spans a vast range of semantic domains, from standard indoor environments (e.g., kitchens, bedrooms) to complex natural landscapes (e.g., forests, deserts) and urban surroundings (e.g., skyscrapers, small houses, historical buildings, parks). Crucially, every sample is rendered at native 4K resolution with complete, pixel-perfect ground truth for metric depth and surface normals. Examples of the dataset are shown in Fig. 5.

B ZüriPano

Outdoor panoramic data collection introduces significant complexities absent in indoor environments, primarily due to dynamic occlusions (e.g., pedestrians, vehicular traffic, and urban activity) and the limited availability of high-resolution, long-range sensing hardware. To date, these challenges have hindered the development of a reliable, LiDAR-based outdoor panoramic evaluation benchmark. To our knowledge, ZüriPano directly addresses this gap by serving as the first real-world outdoor 360° benchmark with survey-grade ground truth.

For acquisition, we employ the Leica RTC360 LiDAR scanner [22], a high-performance reality capture system. The device features an effective operating range of 130 meters and utilizes advanced HDR imaging alongside automated double-scan routines to effectively reject transient moving objects. Our collection encompasses 100 terrestrial laser scans across 11 distinct urban locations in Zürich, Switzerland, capturing a diverse array of architectural styles and open-space geometries. Scans are registered using Leica Cyclone REGISTER 360, with each alignment verified against the exported E57 point cloud. The accompanying panoramic imagery is natively captured at 8K resolution and released at 4K (4096×2048).

During post-processing, we meticulously filter the data to ensure metric reliability. Because the scanner and camera share a single optical center, the dense Euclidean distance rendered from the co-located scan is fundamentally free of cross-view occlusions. The ground truth is stored as 16-bit depth over a $[0, 200]$ m range; this yields a quantization step of ≈ 3 mm, ensuring that storage precision does not limit the scanner's native accuracy. Infinite depth regions (such as the sky) and annotated reflective surfaces (e.g., windows and glass facades, where LiDAR returns are unreliable) are strictly

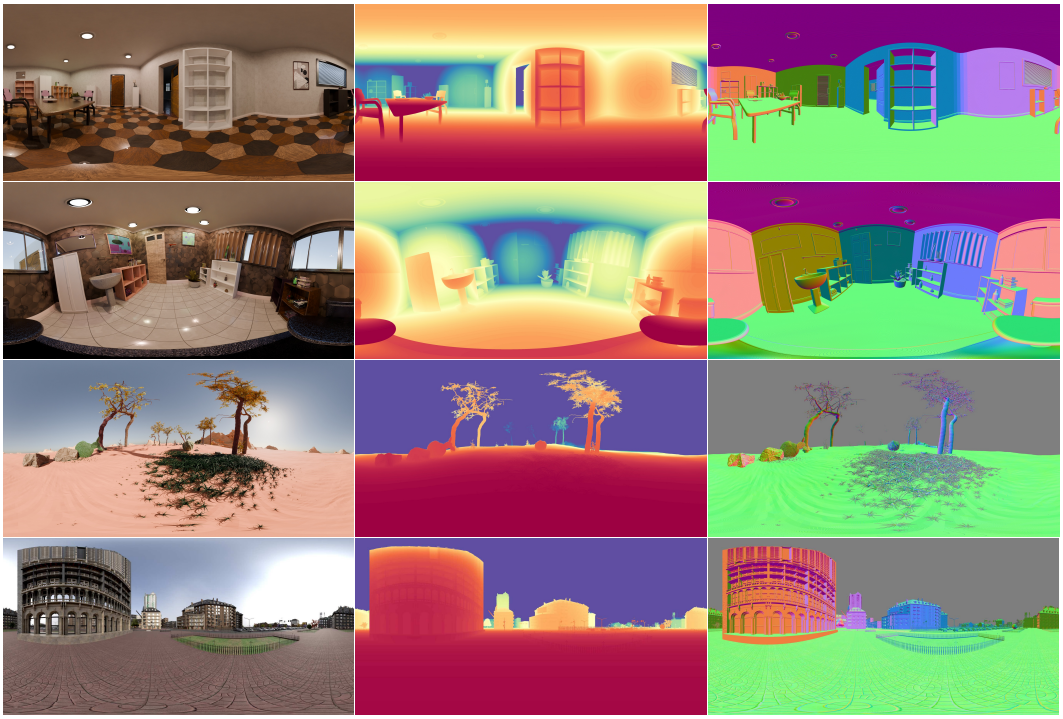

Figure 5: Examples of RGB, depth, and surface normal panoramas from our PanoInfinigen dataset.

masked out. Finally, the validity masks are eroded by 5 pixels around all depth discontinuities. We believe this dataset provides a rigorous testbed for evaluating the robustness of panoramic depth estimation models, specifically regarding long-range accuracy and structural consistency in complex urban environments.

C Additional Related Work Analysis

In this section, we provide further contextualization of our architectural design choices relative to recent methods, expanding upon the discussion in the main text.

Relationship to Recent Multi-Task and Sky-Masking Models. Recent advances in perspective geometry estimation, such as Metric3D v2 [12] and MoGe [47], have demonstrated the benefits of joint depth and normal prediction, as well as the utility of masking unbounded regions. Building on these insights, alongside the DA3 architecture, our framework adopts joint multi-task prediction and explicit sky masking. Our specific architectural contribution in this context lies in efficiency. We demonstrate that delegating unbounded sky regions to a lightweight branch branching off the final depth decoder successfully stabilizes metric regression at a negligible computational cost (as evidenced by the significant RMSE spike when omitted, shown in our ablation studies). Furthermore, regarding joint prediction, we show that a parallel normal decoder initialized from depth weights achieves state-of-the-art panoramic normals with minimal fine-tuning. This indicates that the underlying pre-trained perspective transformer (DA3 ViT) already encodes a geometric prior rich enough to support several modalities simultaneously, without necessitating massive joint retraining from scratch.

Handling Discontinuities: ERP Padding vs. Cubemap Consistency. Some recent approaches to panoramic geometry, such as 360Anything [52] address spatial continuity by employing circular padding (e.g., Circular Latent Encoding) to resolve artifacts at the $\pm 180^{\circ}$ equirectangular (ERP) wrap meridian. However, while ERP-side padding resolves the left-right boundary, it neglects the fact that when multi-face representations (like cubemaps) are used, seams manifest throughout the interior of the unprojected ERP image.

Recent training-free cubemap adaptations attempt to resolve these interior inconsistencies. Most notably, RPG360 [17] converts the panorama to a six-face cubemap and predicts depth for each

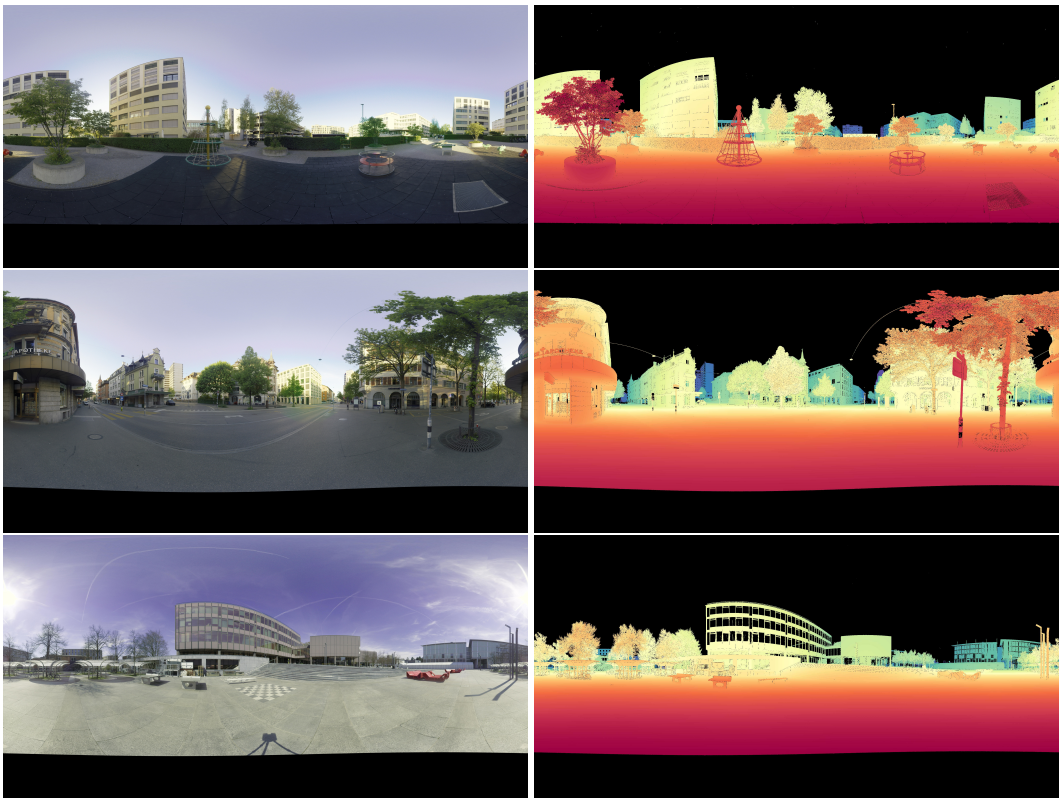

Figure 6: Samples of our ZüriPano dataset.

face independently using a frozen Metric3D v2 model. It then repairs inconsistencies post-hoc via test-time graph optimization over per-face scales and shifts (requiring up to 480 Adam iterations per inference). Because this correction relies on global scalars per face, it fundamentally cannot resolve dense, per-pixel misalignments at the boundaries. In contrast, PaGeR establishes seamless consistency directly during prediction through a joint feed-forward inference pass. By fine-tuning cross-view attention to synchronize features globally and employing cross-face padding to enforce dense local continuity, our method avoids test-time optimization entirely. This approach roughly halves the Absolute Relative error compared to RPG360 across all benchmarks, operating at significantly lower runtime and higher resolution.

D Seam Consistency Metrics

To quantify the geometric consistency of our cubemap projections, we evaluate seam artifacts at three distinct granularities: **Seam Defect Density (SDD)**, **Seam Prevalence (SP)**, and **Seam Severity (SS)**. These metrics monitor depth discontinuities across the $N = 12$ shared face boundaries of the cubemap.

Let $\mathcal{E}$ be the set of all pixel pairs (p, q) that are spatially adjacent across a cubemap boundary, and let $\{E_k\}_{k=1}^{N}$ be the partition of $\mathcal{E}$ into N disjoint sets representing each edge. For any pair $(p, q) \in \mathcal{E}$, we define the log-depth jump as:

$$\Delta_{pq} = |\log \hat{d}_p - \log \hat{d}_q| \tag{4}$$

where $\hat{d}_p$ and $\hat{d}_q$ are the Euclidean ERP linear depths at pixels p and q, respectively.

Seam Defect Density (SDD) measures the global frequency of artifacts by calculating the fraction of total boundary pixels whose depth jump exceeds a tolerance threshold τ:

$$\text{SDD} = \frac{1}{|\mathcal{E}|} \sum_{(p,q)\in\mathcal{E}} \mathbb{I}(\Delta_{pq} > \tau) \tag{5}$$

Seam Prevalence (SP) assesses the distribution of defects across the cubemap structure. An edge E_k is considered prevalent with defects if more than 10% of its constituent pixels exceed τ. The SP metric is the fraction of such "corrupted" edges:

$$\text{SP} = \frac{1}{N} \sum_{k=1}^{N} \mathbb{I}\left(\frac{1}{|E_k|} \sum_{(p,q)\in E_k} \mathbb{I}(\Delta_{pq} > \tau) > 0.1 \right) \tag{6}$$

Seam Severity (SS) captures systemic geometric misalignment by measuring the fraction of edges whose *mean* jump across the entire boundary exceeds a strict magnitude threshold γ:

$$\text{SS} = \frac{1}{N} \sum_{k=1}^{N} \mathbb{I}\left(\frac{1}{|E_k|} \sum_{(p,q)\in E_k} \Delta_{pq} > \gamma \right) \tag{7}$$

where $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the condition is true and 0 otherwise.

In plain words, Seam Defect Density (SDD) measures the local density of pixel-level defects; Seam Prevalence (SP) captures geometric coverage by identifying how many of the twelve edges show visible artifacts; and Seam Severity (SS) identifies systemic collapses by measuring the fraction of edges with high average depth jumps. This triplet allows us to distinguish between widespread, mild jitter and isolated catastrophic failures.

E Performance Trade-offs of Joint vs. Independent Model Training

While our unified panorama estimation model provides the significant advantage of simultaneously outputting scale-invariant (SI) depth, metric depth, surface normals, and sky segmentation from a shared ViT backbone, this multi-task setting inherently requires a compromise in individual task performance. To quantify the impact of joint training and analyze the network's capacity limits, we conducted an ablation study comparing our unified framework against independently trained models.

For this ablation, we trained isolated models for SI depth, surface normals, and a full metric depth model. The independent full metric depth model was trained using dense supervision without depth alignment, thereby directly outputting metric depth.

As expected in multi-task architectures, the results demonstrate that the independent SI depth and surface normal models modestly outperform their counterparts in the unified model. However, the most substantial performance gap is observed between our independent full metric depth model and the metric scale head within the unified setting.

Model	Matterport3D360			Stanford2D3DS			ZüriPano		
	AbsRel↓	RMSE↓	δ_1↑	AbsRel↓	RMSE↓	δ_1↑	AbsRel↓	RMSE↓	δ_1↑
SI Depth									
Unified	9.67	64.69	90.87	5.93	35.34	96.10	9.36	299.61	94.75
Specialized	**9.57**	**63.22**	**91.02**	**5.87**	**34.07**	**96.15**	**9.07**	**274.92**	**95.31**
Metric Depth									
Unified	21.83	123.48	69.50	10.94	**45.43**	90.94	31.97	530.85	39.30
Specialized	**17.71**	**91.27**	**86.56**	**10.01**	46.28	**92.07**	**18.19**	**335.90**	**77.90**

(a) Depth Estimation (SI and Metric)

Model	Structured3D			
	Mean↓	MSE↓	5°↑	22.5°↑
Surf. Normals				
Unified	5.49	174.9	79.91	92.83
Specialized	**4.96**	**157.4**	**82.18**	**93.53**

(b) Surface Normals

Table 7: **Unified vs. Specialized Models.** We compare our multi-task unified PaGeR model against individually trained specialized models for each modality. **(a)** Evaluation of Scale-Invariant (SI) and Metric depth across three datasets. **(b)** Surface normal estimation evaluated on the Structured3D dataset. Top-performing metrics for each modality pair are highlighted in **bold**.

This discrepancy is directly tied to gradient flow and backbone freezing constraints. For the independent full metric model, the ViT backbone is fully unlocked, allowing the network to extract metric-specific cues early in the encoder stages. This deep integration greatly enhances metric depth accuracy and effectively obviates the need for separate, heavy metric heads.

Conversely, in the unified model, the ViT backbone must remain frozen with respect to the metric scale head. We found that allowing gradients from the metric scale head to flow back into the shared ViT during joint training fundamentally conflicts with the optimization of the other tasks, leading to a severe degradation in the quality of both SI depth and surface normal predictions. Thus, keeping the ViT frozen preserves the integrity of the SI depth and normals but creates a bottleneck for metric depth performance.

These findings suggest that while our shared representation is highly efficient, it is currently capacity-constrained when balancing the extraction of both scale-invariant and purely metric cues. A careful adaptation of the ViT backbone—such as introducing task-specific routing or slightly expanding capacity to better accommodate these competing gradients—represents a highly promising direction for future work.

F Loss Weights and Hyperparameters

To facilitate full reproducibility and provide transparency regarding our training setup, we detail the complete configuration and computational infrastructure of our framework. This includes scaling factors for the scale-invariant depth, metric scale, surface normal, and sky segmentation heads. For clarity and ease of reference, we present the precise loss weights λ, optimization schedules, learning rates, and architectural choices across all modalities in Table 8.

Modality	Loss Term	Symbol	Weight
Depth	Confidence L1 Loss	λ_{L1}	1.0
	Lambda Confidence	λ_c	0.2
	Gradient Loss	λ_{grad}	40.0
	Normals Consistency Loss	λ_{norm}	0.6
Normals	Cosine Similarity Loss	λ_{cos}	1.0
	Perceptual Loss	λ_{perc}	0.5
Metric Scale	Confidence L1 Loss	λ_{L1}	1.0
	Lambda Confidence	λ_c	0.2
	DPT Subsampling Factor	F	4.0
Sky Segmentation	Binary Cross-Entropy Loss	λ_{BCE}	1.0
	Focal Loss	λ_{Focal}	0.4
	Dice Loss	λ_{Dice}	1.0

Table 8: Hyperparameter configuration for our default model.

G Evaluation of Alternative DA3 Input Configurations

To ensure a fair and rigorous comparison against the vanilla DA3 baseline presented in Table 4a of the main text, we investigated whether alternative multi-view projection strategies could yield better zero-shot panoramic performance. While our framework relies on a non-overlapping 6×90° cubemap, it is a reasonable hypothesis that a pre-trained multi-view model like DA3 might benefit from explicit overlapping regions, denser view sampling, or independent single-view processing.

To test this, we implemented and evaluated several alternative input configurations for the vanilla DA3 model on the Stanford2D3DS scale-invariant depth benchmark:

- **(a) 6×110° overlapping cubemap (joint):** The standard six cubemap faces widened to a 110° field of view (FoV), providing approximately 20° of overlap between adjacent views.
- **(b) 12×90° icosahedral (joint):** A twelve-view arrangement distributed across an icosahedron, mirroring the official panorama processing pipeline of recent methods like MoGe.
- **(c) 20×70° icosahedral (joint):** A denser, smaller-FoV configuration using tangent views at the centers of icosahedron faces, resulting in roughly 28° of overlap. This configuration processes the views jointly but consumes approximately 3.3× the pixel budget of our standard cubemap.
- **(d) 20×70° independent:** Using the same dense 20-view configuration as (c), but predicting depth for each view in complete isolation. The final panorama is reconciled by solving for per-view scale factors via a global least-squares optimization over the overlap graph.

As shown in Table 9, despite the intuition that explicit overlaps or denser view sampling might aid the pre-trained perspective model, the standard non-overlapping 6×90° cubemap achieves the best quantitative results among all vanilla DA3 configurations.

Furthermore, to ensure that the stitching mechanism was not bottlenecking the performance of the overlapping configurations, we tested four distinct compositing rules on the best-performing alternative (configuration c): feathered blending, uniform mean, nearest view selection, and per-pixel median. Across all four compositing strategies, the variance in AbsRel was ≤ 0.24. Because the

DA3 Input Configuration	AbsRel ↓	δ_1 ↑
6×90° cubemap, joint (Table 4a)	**6.58**	**95.94**
(a) 6×110° overlapping cubemap, joint	9.37	93.58
(b) 12×90° icosahedral, joint	8.33	94.58
(c) 20×70° icosahedral, joint	7.65	94.95
(d) 20×70° independent, LS-aligned stitch	9.45	92.72
PaGeR (Ours, reference)	5.93	96.10

Table 9: **Comparison of alternative DA3 input configurations on the Stanford2D3DS dataset (scale-invariant depth).** Metrics are scaled by ×100. Contrary to expectations, the non-overlapping 6×90° cubemap provides the strongest baseline performance for the vanilla model.

consensus choice has a negligible impact on overall performance, the standard 6×90° cubemap remains the most robust and computationally efficient baseline against which to evaluate our proposed adaptations.

H Analysis of Multi-View Consistency

To comprehensively evaluate the robustness of our representations, we conducted an additional ablation to assess the multi-view geometric consistency of our framework relative to prior state-of-the-art methods. Specifically, we investigate whether the model predicts geometrically compatible 3D outputs for two distinct panoramic frames of the same scene captured with a small camera offset.

Evaluation Protocol. Given two panoramic frames (Frame A and Frame B) of a scene from nearby viewpoints with known relative poses, we predict the depth for each frame independently. Following our standard scale-invariant evaluation protocol, we scale-align each prediction to its respective ground truth. We then unproject the predicted depths into 3D point clouds and place both reconstructions into a shared world coordinate frame.

We quantify multi-view consistency using two complementary metrics:

- **Inter-view Reprojection AbsRel:** This metric projects each predicted 3D point from Frame A into the camera view of Frame B. It then computes the Absolute Relative Error between the projected range of Point A and the actual depth predicted by Frame B along that specific ray.

- **Co-visible Chamfer Distance (% of scene depth):** A correspondence-free nearest-neighbor distance metric computed between the two overlapping point clouds. This metric is particularly sensitive to global misalignments, such as scale drift or systemic warping between the two independent predictions.

Results. We evaluated this consistency on a zero-shot basis using 40 real-world panorama pairs extracted from 5 test buildings within the Matterport3D dataset (with baseline distances ranging from 0.30 to 0.97 meters).

As demonstrated in Table 10, PaGeR achieves the highest multi-view consistency among all compared methods. It significantly outperforms optimization-based approaches like RPG360 and continuous ERP models like DA^2. This confirms that the strong perspective priors preserved by our cubemap architecture not only yield highly accurate single-view geometry but also naturally result in robust, geometrically stable structures across novel viewpoints without requiring multi-view test-time optimization.

Method	Reproj. AbsRel (%) ↓	Chamfer (% depth) ↓
GT floor (Reference)	0.07	0.95
UniK3D	58.34	4.05
PanDA	25.78	8.15
DA^2	3.98	3.05
RPG360	3.71	3.71
PaGeR (Ours)	**2.48**	**2.26**

Table 10: **Multi-view consistency on real Matterport3D panorama pairs (Zero-shot).** Both metrics evaluate the alignment of two independent single-view predictions when unprojected into a shared 3D space.

I Computational Efficiency and Resource Benchmark

To evaluate the practical utility and scalability of PaGeR, we benchmark its computational footprint against other baselines. Table 11 provides a comprehensive comparison across four key performance dimensions: inference runtime, peak memory consumption during evaluation, native processing resolution, and total training data requirements.

Method name	Runtime [s]	Peak Memory [GB]	Resolution [px]	Data size [10^3]
DreamCube	6.06	15.2	1024×2048	N/A
DepthAnyCamera	0.11	0.3	512×1024	800
MoGe	36.05	3.6	1024×2048	8860
DAP	0.10	3.7	512×1024	1700
RPG360	5.83	13.2	512×1024	N/A
EGformer	0.31	1.2	512×1024	70
UniK3D	0.41	3.3	560×1106	700
PanDA	0.63	3.5	504×1008	120
DA^2	0.05	2.8	546×1092	606
Ours	0.48	12.8	1008×2016	100

Table 11: Comparison of various methods based on performance metrics.

J Additional Quantitative Results

	Matterport3D360			Stanford2D3DS		
Method	AbsRel ↓	RMSE ↓	δ_1 ↑	AbsRel ↓	RMSE ↓	δ_1 ↑
DAP [28]	13.44	43.36	84.90	8.08	25.57	93.97
RPG360 [17]	11.79	38.78	87.83	10.50	29.74	89.33
UniK3D† [33]	13.12	<u>38.36</u>	86.69	9.25	26.18	93.83
PanDA* [4]	<u>10.64</u>	38.71	<u>88.39</u>	7.92	23.18	94.97
DA^2 [24]	10.81	38.38	87.01	<u>6.99</u>	<u>23.00</u>	<u>95.95</u>
PaGeR (Ours)	**9.95**	**37.90**	**88.73**	**5.33**	**21.93**	**96.06**

Table 12: **Quantitative comparison between PaGeR and state-of-the-art baselines on Matterport3D360 and Stanford2D3DS in narrow-depth setting ($[0, 5]$ m).** Best and second-best results are indicated in **bold** and <u>underlined</u>, respectively. Methods optimized using in-domain training are marked with †, and affine-invariant methods are denoted with *.

	Matterport3D360			Stanford2D3DS			ZüriPano		
Method	AbsRel ↓	RMSE ↓	δ_1 ↑	AbsRel ↓	RMSE ↓	δ_1 ↑	AbsRel ↓	RMSE ↓	δ_1 ↑
DAP [28]	15.76	85.04	82.61	9.76	46.84	92.29	19.83	570.90	73.01
RPG360 [17]	15.41	79.65	82.39	11.92	47.08	87.71	<u>18.22</u>	<u>457.16</u>	<u>78.59</u>
UniK3D† [33]	14.82	69.20	83.69	9.92	40.69	93.32	28.09	794.97	46.52
PanDA* [4]	13.83	82.87	84.20	10.69	55.80	90.70	25.05	507.48	56.83
DA^2 [24]	<u>11.07</u>	<u>67.77</u>	<u>89.16</u>	<u>7.68</u>	<u>37.50</u>	<u>95.57</u>	60.53	859.89	2.22
PaGeR (Ours)	**9.69**	**64.86**	**90.85**	**6.01**	**35.96**	**96.02**	**9.47**	**305.08**	**94.62**

Table 13: **Quantitative comparison between PaGeR and state-of-the-art baselines across Matterport3D360, Stanford2D3DS, and ZüriPano on downscaled 512×1024 resolution.** Best and second-best results are indicated in **bold** and <u>underlined</u>, respectively. Methods optimized using in-domain training are marked with †, and affine-invariant methods are denoted with *.

K Additional Qualitative Examples

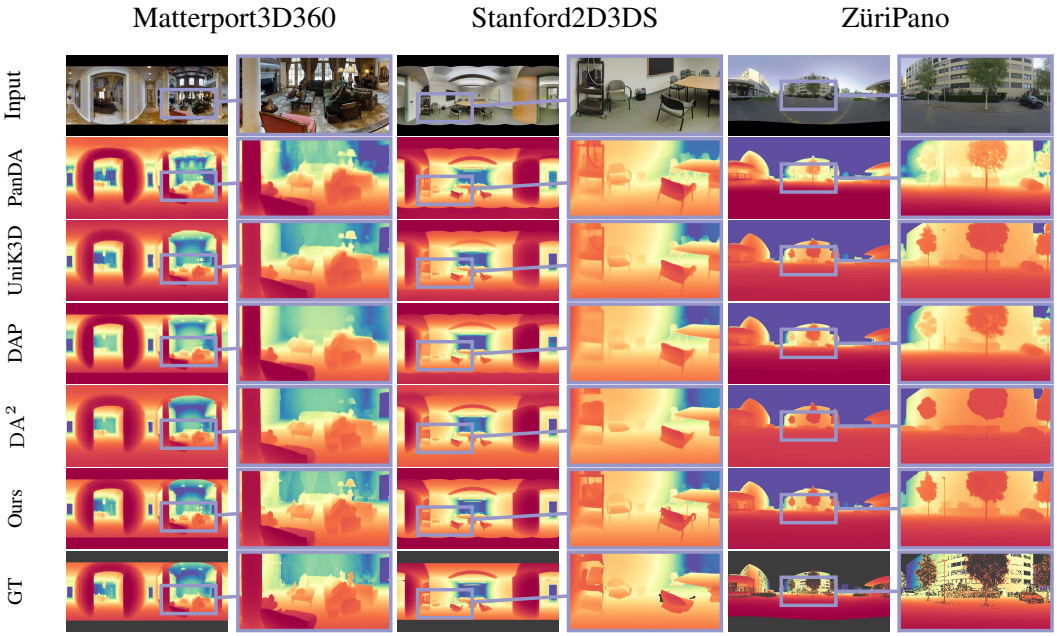

Figure 7: **Qualitative comparison of panoramic depth estimation.**. Visual results from PaGeR, compared to the subset of the best evaluated baselines, shown alongside the RGB input and ground-truth depth on Matterport3D360, Stanford2D3DS, and ZüriPano panoramas. (Best viewed zoomed in.)

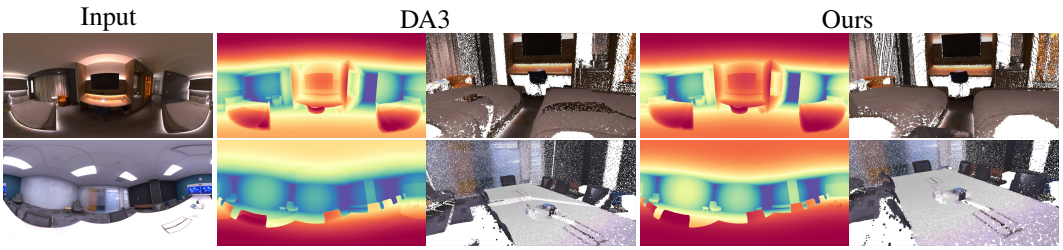

Figure 8: Comparison to vanilla DA3.

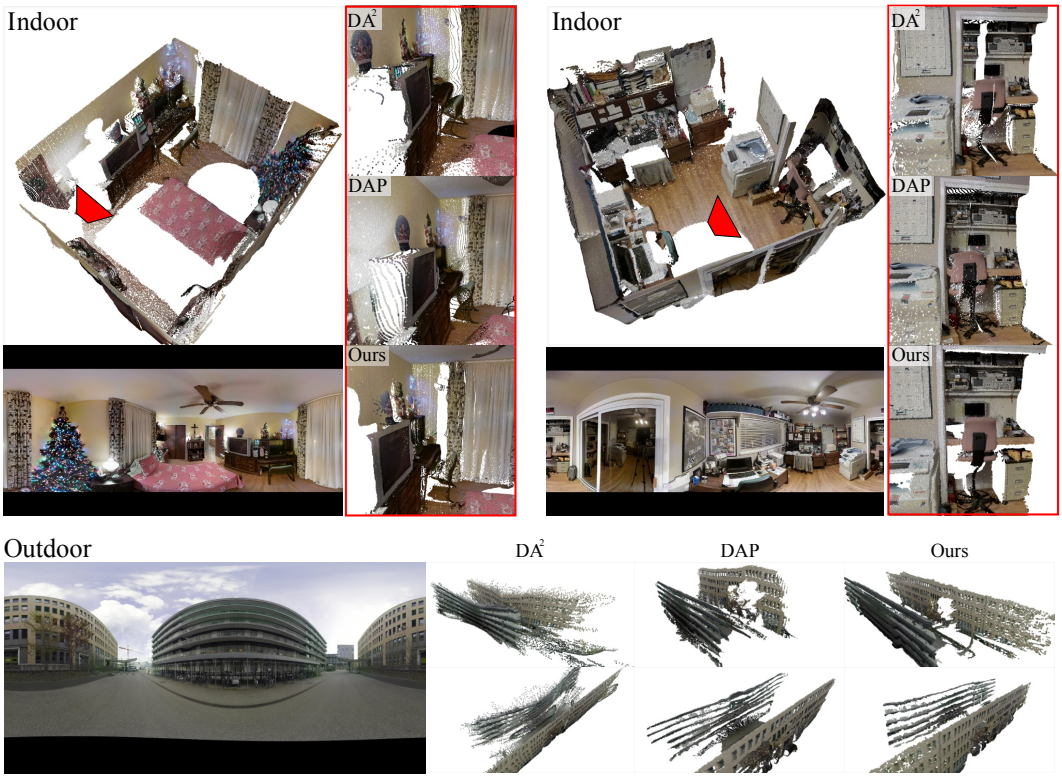

Figure 9: **Qualitative point-cloud comparison.** Indoor scenes (top) and an outdoor scene (bottom) are rendered as point clouds alongside the corresponding panoramic input images for competitors and our method. For the indoor examples, we show our point cloud reconstruction with zoomed-in novel-view rendering comparison to the main competitors, highlighted by red boxes.



Figure 10: Examples of measured distances in our predicted point cloud. The measures are given in meters.

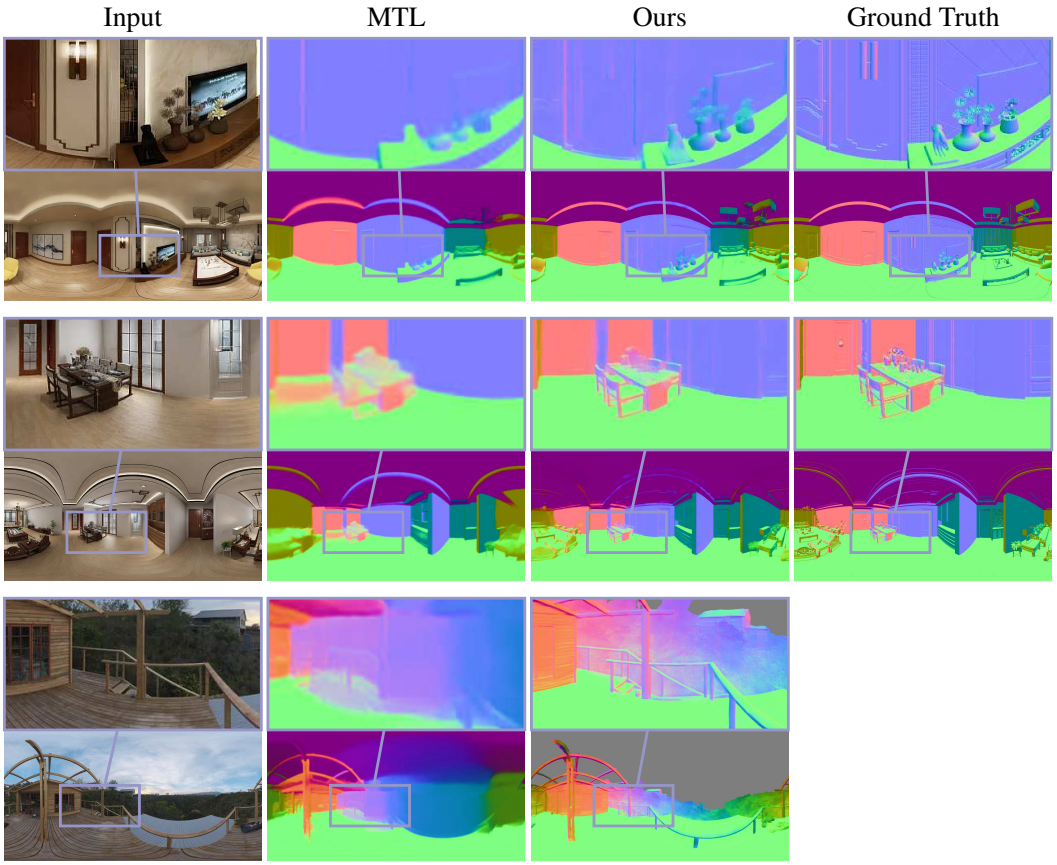

Figure 11: **Qualitative comparison of panoramic surface normals estimation.** Visual results from PaGeR and MTL (best available baseline method), shown alongside the RGB input and ground-truth depth on panoramas from the Structured3D dataset. (Best viewed zoomed in.)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper clearly presents the topic, scope, and achieved results in the abstract and introduction, which are later solidified in the thorough explanations of the method section and backed up by the extensive experiments and ablations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Paper clearly discusses the limitations of our work in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper thoroughly explains the methodology of our work, including architectures, datasets, optimization and training settings and computational demands in order to maximize the reproducibility of our work.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: To further strengthen the reproducibility of our approach, we intend to open-source model, our own generated and collected datasets, as well as the code, for complete reproducibility of achieved results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: In the Section 3, we fully disclose the train data, as well as optimization parameters and training procedure.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We don't utilize the error bars in our experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, provided in Section 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We fully conform to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available data and models, while all of the related work is properly cited throughout the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All of the models, code, and the data will be released and thoroughly documented.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.