Curvature-Aware Pointmap Regression for 3D Reconstruction

Taro Iyadomi[1] Jiayi Liwei[2] Max Bregler[2] William Chang[2]

[1]Department of Statistics and Data Science, Northwestern University
taroiyadomi@u.northwestern.edu

[2]Department of Mathematics, University of California, Los Angeles
{jiayi2024, maxbregler, chang314}@g.ucla.edu

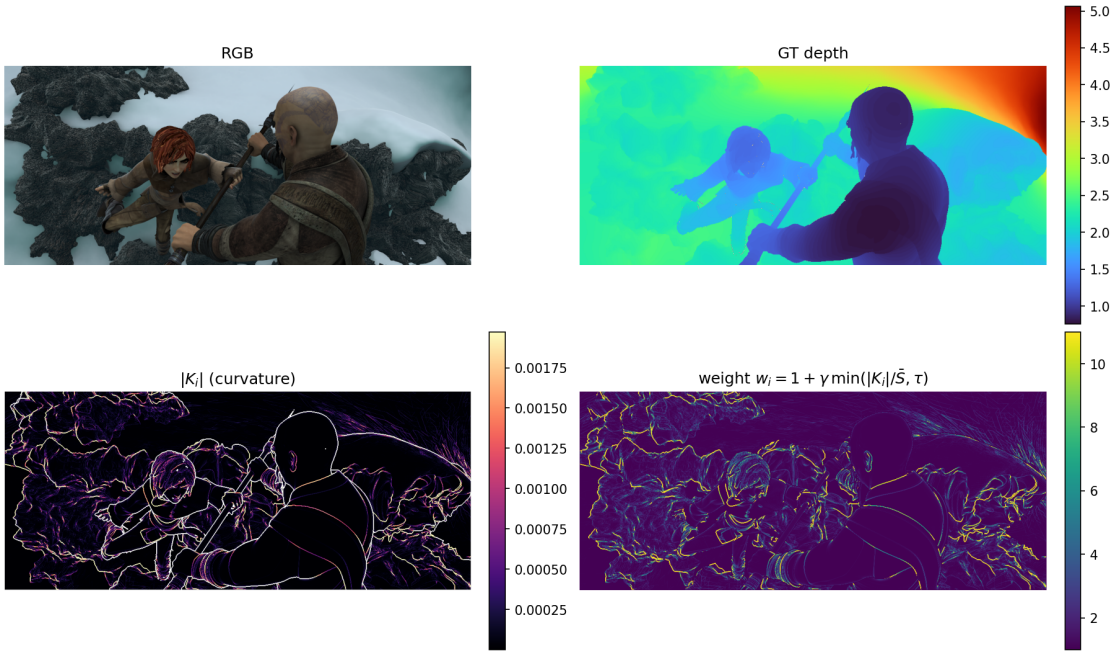


Figure 1: Curvature is a free, pixel-aligned supervisory signal. On a Sintel frame (`ambush_6`), we compute the mean-curvature magnitude $|K_i|$ of the ground-truth pointmap (Eq. (2)) and the per-pixel loss weight $w_i = 1 + \gamma \min(|K_i|/\bar{S}, \tau)$ at $\gamma = 1$ from the input image and ground-truth depth. The curvature signal and the reweighting are concentrated at object boundaries and surface creases where the reconstruction error is largest.

Abstract

Feed-forward pointmap regression, introduced by DUSt3R and extended to dynamic scenes by D^{2}USt3R, has become a leading paradigm for 3D reconstruction, yet existing methods typically only focus on the positional values of the pixels they train on. These methods neglect the inherent geometric properties of objects living in 3D spaces, saying nothing about how surfaces curve or fold. This missing cue, discrete curvature, comes at low cost. It is a deterministic, pixel-aligned function of the ground-truth pointmaps, computable at each pixel with no extra data and minimal computational overhead. To leverage the geometric signal discrete curvature provides, we introduce a curvature-weighted position loss, a reweighting of the DUSt3R regression that emphasizes high-curvature regions, introducing no new learnable parameters and reducing to the standard regression loss when its strength hyperparameter is set to zero. We finetune from the publicly released D^{2}USt3R checkpoint, a pointmap regressor pretrained for dynamic

scenes, and show that in a matched five-seed study, curvature weighting lowers depth error on seven of the eight metrics we track, with notable improvements in zero-shot evaluation. We perform a weight-strength sweep and report that zero-shot error decreases monotonically as we strengthen the curvature signal.

Keywords: pointmap regression · 3D reconstruction · discrete curvature · geometric supervision · depth estimation

1 Introduction

Recovering 3D geometry from images is a long-standing problem in computer vision. A recent and influential shift has replaced multi-stage Structure-from-Motion and Multi-View Stereo pipelines with direct, feed-forward *pointmap regression*. DUSt3R [51] predicts, for each pixel of an image pair, a 3D point expressed in a common camera frame, folding depth, relative pose, and dense correspondence into a single pixel-aligned representation $X \in \mathbb{R}^{W\times H\times 3}$ that is learned by regressing against ground-truth pointmaps. By collapsing a long pipeline into one regression target, this formulation sidesteps the error accumulation of classical methods and has become the backbone for a growing family of reconstruction models.

The original formulation, however, assumes a static scene and its performance degrades when objects within the scene move. Alignment driven by camera pose alone cannot reconcile pixels whose 3D position changes across frames. MonST3R [55] mitigates this by fine-tuning on dynamic video, but still applies a single rigid transform per frame, so it cannot represent independently moving points. D^2USt3R [19] closes much of this gap with *static–dynamic aligned pointmaps*: static pixels are warped by camera pose and dynamic pixels by optical flow, supervised through separate static and dynamic alignment losses that are localized by occlusion and dynamic masks. The result is state-of-the-art depth, pose, and correspondence estimation in dynamic scenes, with the encoder frozen and only the decoder and prediction heads fine-tuned.

Despite these advances, the supervision underlying all of these models is fundamentally a position loss, a per-pixel Euclidean distance between predicted and ground-truth 3D points, optionally reweighted by confidence and restricted to valid regions (cf. Eq. (3)). Such a loss anchors where each point lies but says nothing about how the surface bends from a pixel to its neighbors. Two consequences follow. First, because the penalty grows with coordinate magnitude, the objective is dominated by large-scale structure and underweights fine local geometry. Second, reconstruction error concentrates where surface shape varies fastest, at object boundaries, corners, occlusion contours, and the silhouettes of moving objects, the regions that govern the perceived quality of a dynamic reconstruction. Geometric cues such as surface normals and curvature have long been used to sharpen monocular and multi-view reconstruction [1, 3, 36, 37, 46, 54], yet current pointmap regressors leave this signal unused.

Our central observation is that, in the pointmap setting, this signal is free. Discrete curvature is a deterministic, pixel-aligned function of the ground-truth pointmap itself, the regression target that pointmap methods build from dataset depth, intrinsics, and poses. Because the pointmap is pixel-aligned, a pixel's image neighbors $(u \pm 1, v \pm 1)$ are also its neighbors on the surface, so a per-pixel mean-curvature magnitude map $K \in \mathbb{R}^{W\times H}$ can be computed analytically during training at negligible computational cost. To do this, we apply the discrete Laplace–Beltrami operator to the coordinate map to obtain the mean-curvature normal $\Delta_{\mathcal{M}}\mathbf{x} = -2H\mathbf{n}$ [30], whose magnitude measures how fast the surface bends and which we make precise in our preliminaries (Eq. (2)).

Building on this observation, we augment the pointmap position regression with an explicit curvature signal in a single, minimal form, a *curvature-weighted position loss* that reweights the

position residual at pixel i by a factor $w_i = 1 + \gamma\,|K_i|$ that grows with local curvature (normalized across the image and clamped; Sec. 4.2). The construction introduces no new parameters and adds a single strength hyperparameter γ. We freeze the encoder of the released D^{2}USt3R checkpoint and finetune the decoder and dense prediction transformer (DPT) heads with our curvature-weighted loss on PointOdyssey and TartanAir. Setting $\gamma = 0$ recovers the standard confidence-weighted regression loss from DUSt3R. Discrete curvature is numerically delicate (large at silhouettes, undefined at image borders, and heavy-tailed), so the weighted loss restricts to valid pixels, zeroes borders, normalizes across each image, and clamps outliers (Sec. 4.2).

In summary, we make the following contributions:

- **A parameter-free curvature-aware objective.** We identify a blind spot in pointmap regression. The position loss constrains point coordinates but leaves the local surface shape unsupervised. We show that the missing cue, discrete curvature, is latent in the ground-truth pointmaps the DUSt3R family constructs. Grounded in the Laplace–Beltrami/mean-curvature relation, this geometric cue can be converted into our curvature-weighted position loss that redistributes existing supervision toward higher-curvature regions, adds no learnable parameters and a single hyperparameter γ, and recovers the standard loss at $\gamma = 0$.
- **A controlled evaluation of the effect.** In a matched five-seed study with a first-order gradient control and a weight-strength sweep, curvature weighting lowers depth error on seven of the eight metrics we track, with the largest and most consistent gains in the zero-shot regime and a monotonic zero-shot trend.

2 Related Work

Pointmap-Based 3D Reconstruction. Classical 3D reconstruction pipelines decompose the problem into stages: Structure-from-Motion recovers sparse structure and camera poses [42], dense Multi-View Stereo (MVS) fills in per-pixel depth [13], and a surface extraction step produces the final mesh. Learned MVS methods such as MVSNet [53] replace hand-crafted matching with cost-volume regression but retain the multi-stage decomposition and its attendant error accumulation. DUSt3R [51] introduced a fundamentally different paradigm: feed-forward regression of pixel-aligned pointmaps as a unified surrogate for stereo matching, depth, and pose. MASt3R [25] extends this with a local-feature matching head that improves correspondences. More recent feed-forward models scale this paradigm to many views in a single pass: VGGT [49] jointly regresses pointmaps, depth, and camera parameters across multiple views without the test-time global alignment required by DUSt3R. Follow-up work adapts the paradigm to dynamic scenes: MonST3R [55] fine-tunes on moving-scene video, and D^{2}USt3R [19] introduces static–dynamic aligned pointmaps that separately supervise rigid and non-rigid regions. We build on the DUSt3R pointmap regression paradigm and initialize from the released D^{2}USt3R checkpoint, adding geometric supervision to the position loss rather than modifying the representation or alignment procedure. Its dynamic-scene training objective was not released, so our baseline ($\gamma = 0$) finetunes the checkpoint with the standard confidence-weighted position regression.

Neural and Dynamic Scene Representations. An alternative to explicit pointmap regression is to encode a scene in a continuous neural field. NeRF [31] parametrizes radiance and density with a multilayer perceptron and renders novel views via volumetric ray-marching, while 3D Gaussian Splatting [23] replaces the implicit field with an explicit set of anisotropic Gaussians that are rasterized differentiably, enabling real-time rendering. Extending such methods to dynamic scenes

introduces additional complexity: D-NeRF [35] learns a deformation field that warps a canonical volume over time, and Nerfies [34] models non-rigid deformations through per-frame latent codes. Dense simultaneous localization and mapping (SLAM) systems such as KinectFusion [33] fuse depth maps into volumetric representations in real time but assume rigid scenes and active depth sensing. These methods achieve high geometric detail through per-scene optimization, fitting each scene from scratch, whereas pointmap regressors target feed-forward, scene-agnostic inference. We situate our contribution in this feed-forward regime, where reconstruction quality is shaped by the training supervision rather than by a per-scene optimization prior.

Geometric Cues and Geometric Supervision. Surface normals and curvature have long served as geometric priors for shape recovery. Curvature constraints have been used for recovering geometry from a single image [46], while a large body of work predicts surface normals to improve depth estimation and reconstruction [1, 3, 11, 12, 37]. Dharmasiri *et al.* [9] regress surface curvature directly from a single RGB image with a convolutional neural network, treating it as an auxiliary task that in turn improves depth and normal prediction. Methods that explicitly couple depth and surface normals, such as GeoNet [36], enforce a consistency between depth and normals within a single, static frame, again through a first-order quantity. Large-scale multi-task learning [10] treats normal prediction as one of several jointly trained geometric tasks, and adaptive normal-consistency constraints [29] reweight the normal supervision by local reliability. Similar ideas have been adopted in neural implicit reconstruction, including MonoSDF [54], Ref-NeRF [48], IGR [16], NeuS [50], and Manhattan-world constrained reconstruction [17]. With the exception of Dharmasiri *et al.* [9], these methods use first-order geometric information through normals or depth gradients. Our focus is on curvature, a second-order quantity that peaks at boundaries, creases, and thin structures. Unlike prior work, which predicts normals or curvature as an output target, we use this signal as a weight on an existing position loss, redistributing supervision toward high-error regions without changing what is supervised.

Depth Regularization and Boundary-Aware Losses. A common strategy for improving depth reconstruction is to modulate smoothness according to image content. Monodepth [14] introduced edge-aware smoothness penalties, later refined in Monodepth2 [15]. MiDaS [39] employs multi-scale gradient matching, while stereo approaches such as PM-Huber [20] relax regularization near image edges. Higher-order priors including Total Generalized Variation [4] and variational stereo formulations [24] encourage piecewise-planar structure, while hierarchical conditional random fields [27] and DPT-style architectures [38] improve multi-scale reasoning. These methods encode generic smoothness assumptions but do not explicitly supervise local surface bending, the quantity our curvature signal targets.

Discrete Differential Geometry and Laplacian-Based Methods. The relationship between curvature and the Laplace–Beltrami operator is well established in geometry processing. Discrete differential-geometry operators [30, 41] provide practical estimators of curvature on meshes and point clouds, while graph-Laplacian constructions converge to their continuous counterparts under suitable sampling assumptions [2, 6]. Laplacian operators have been used extensively for mesh smoothing and editing [8, 32, 44, 47], spectral shape analysis [7, 18, 21, 26, 40, 45], and Poisson surface reconstruction [22]. In contrast to these works, we use the Laplacian only as a local operator for extracting curvature from pointmaps. Discrete-Laplacian curvature also regularizes neural implicit surfaces [28], and pointmap-derived boundary smoothness regularizes feed-forward Gaussian splatting [43], but neither targets the pointmap regression loss itself. To our knowledge, curvature-derived supervision has not

been explored in feed-forward pointmap regression for dynamic scene reconstruction.

3 Preliminaries

3.1 Pointmaps and Pairwise Regression

Let I^1, I^2 be an image pair of resolution $W \times H$. A pointmap $X^{n,m} \in \mathbb{R}^{W\times H\times 3}$ gives, per pixel of image n, its 3D location in the frame of camera m. The network of the DUSt3R family [19, 51, 55] regresses $X^{1,1}, X^{2,1}$ with confidences $C^{1,1}, C^{2,1}$, from which depth, relative pose, and dense correspondence follow. It is trained by a per-pixel confidence-weighted regression of the predicted pointmaps against the ground-truth pointmaps (Eq. (3)). Our method reweights this pairwise loss and leaves the network, the pointmap representation, and the multi-view global-alignment stage that fuses per-pair predictions for $N > 2$ views unchanged.

3.2 Ground-Truth Pointmap Construction

The ground-truth pointmaps are computed from the per-frame depth, intrinsics, and camera poses the datasets supply. Given world-to-camera poses $P^n, P^m \in \mathbb{R}^{4\times 4}$, intrinsics K_n, and a depth map $D_n \in \mathbb{R}^{W\times H}$, the ground-truth pointmap of image n expressed in camera m is

$$\bar{X}^{n,m} = P^m \, (P^n)^{-1} \, h\big(K_n^{-1} D_n\big), \qquad h : (x, y, z) \mapsto (x, y, z, 1), \tag{1}$$

so that, for a pair, the targets $\bar{X}^{1,1}, \bar{X}^{2,1}$ both live in the coordinate frame of the first camera [19]. The second view is brought into this frame by the single rigid transform $P^m(P^n)^{-1}$, applied identically to every pixel. This assumes a static scene, so the target is correct for static points but misplaced for points that move between frames. D^{2}USt3R addresses this with a static/dynamic split that re-targets moving pixels through optical flow; our baseline instead supervises every pixel against the rigid target (Eq. (3)). Curvature is read from this ground-truth pointmap independently of the split.

3.3 Mean Curvature from the Pointmap Laplacian

The method relies on one fact from discrete differential geometry. Applying the Laplace–Beltrami operator of a surface $\mathcal{M}$ to its own coordinate map returns the mean-curvature normal,

$$\Delta_{\mathcal{M}} \, \mathbf{x} = -2H \, \mathbf{n}, \qquad (L\chi)_n(p) \;\approx\; -\,2H_n(p)\, \mathbf{n}_n(p), \tag{2}$$

where H is the mean curvature and $\mathbf{n}$ the unit surface normal, so its magnitude $\|\Delta_{\mathcal{M}}\mathbf{x}\| = 2|H|$ measures how fast the surface bends.[1] Because a pointmap is pixel-aligned, a pixel's image neighbors are also its surface neighbors, so $\Delta_{\mathcal{M}}$ is well approximated by a local stencil over them, the four-neighbor umbrella Laplacian L we use in Sec. 4.2 [30]. Curvature is thus a local, per-pixel function of the pointmap, reading only the immediate neighborhood with no global solve.

Mean curvature describes how a surface bends, a property of the object rather than of its pose, so a rigid or articulated part bends the same way whether or not it is moving. The quantity in Eq. (2) should therefore agree across frames even in the dynamic regions where the static construction of Eq. (1) breaks down. Curvature is thus a geometric cue complementary to optical flow, with flow constraining where a dynamic point goes and curvature constraining what shape it keeps once it gets there.

[1] Up to the sign and normalization convention of $\Delta_{\mathcal{M}}$ and H.

Discretization Choice. Computing K_i requires a discrete Laplacian stencil over a pixel's four image neighbors and two estimators to be available. On a triangle mesh the standard estimator of Δ_M is the cotangent Laplacian [30], which weights each neighbor by cotangents of the opposing angles and is exact on planar triangles. A pointmap is not an arbitrary mesh. Its connectivity is the fixed image grid, and after `avg_dis` normalization the incident triangles formed by a pixel and its four image neighbors have nearly uniform angular weights except at depth discontinuities, where the cotangent formula is itself dominated by degenerate triangles at silhouettes. The four-neighbor umbrella Laplacian [30] $(L\chi)_i = \frac{1}{4}\sum_{n\in\mathcal{N}(i)} \bar{X}_n - \bar{X}_i$ therefore approximates $\Delta_M\chi$ up to a smooth reweighting that is absorbed into the tuning of γ and the clamp τ, while remaining numerically well-defined at silhouettes where the cotangent estimator diverges. This trade, an approximation constant absorbed into a scaling hyperparameter in exchange for stability at the pixels we most want to upweight, is the reason we use the umbrella form throughout.

4 Methods

4.1 Baseline Finetuning Objective

For an image pair, the D^{2}USt3R checkpoint predicts, for each view, a pointmap in the reference (first view) camera frame, together with a per-pixel confidence. We finetune it by confidence-weighted regression of the predicted pointmap against the ground-truth pointmap, keeping the encoder frozen and training the decoder and DPT heads. Let $\hat{X}_i$ and $\bar{X}_i$ be the predicted and ground-truth 3D points at pixel i, both normalized by the average point-to-origin distance over the valid set (the `avg_dis` convention, making the objective scale-invariant). Let M_v be the valid-pixel mask of view v, $c_i > 0$ the predicted confidence, and $\alpha = 0.2$. The baseline objective, summed over the two views $v \in \{1, 2\}$, is

$$\mathcal{L}_{\text{base}} = \sum_v \frac{1}{|M_v|} \sum_{i\in M_v} \Big[\, c_i\, \ell_i \;-\; \alpha \log c_i \,\Big], \qquad \ell_i = \big\|\hat{X}_i - \bar{X}_i\big\|_2. \tag{3}$$

We finetune the released D^{2}USt3R checkpoint with this objective and apply the curvature weight to the residual ℓ_i. The matched $\gamma = 0$ versus $\gamma = 1$ comparison then isolates the effect of the weight.

4.2 Curvature-Weighted Supervision

Our contribution reweights the pointwise position residual ℓ_i by a local geometric signal of the ground-truth surface, concentrating supervision on geometrically complex regions (edges and corners), where reconstruction error concentrates (Fig. 1):

$$\mathcal{L} = \sum_v \frac{1}{|M_v|} \sum_{i\in M_v} \Big[\, c_i \left(w_i\, \ell_i\right) \;-\; \alpha \log c_i \,\Big], \qquad w_i = 1 + \gamma\, \min\!\left(\frac{S_i}{\bar{S}},\, \tau\right). \tag{4}$$

Here $\bar{S} = \frac{1}{|M_v|}\sum_{j\in M_v} S_j$ is the mean signal over the image, so that γ is dimensionless and comparable across signals and scenes. The weight multiplies only the residual, not the confidence term $-\alpha \log c_i$. Setting $\gamma = 0$ yields $w_i \equiv 1$ and recovers $\mathcal{L}_{\text{base}}$, so the baseline and our method share a single code path and the comparison is matched.

Geometric Signal. The signal S_i is the magnitude of the discrete mean curvature of the ground-truth pointmap, estimated by the umbrella Laplacian over the four image neighbors $\mathcal{N}(i)$,

$$S_i = |K_i| = \tfrac{1}{2}\Big\| \tfrac{1}{4} \sum_{n\in\mathcal{N}(i)} \bar{X}_n \;-\; \bar{X}_i \Big\|. \tag{5}$$

The signal K_i is a deterministic function of the same ground-truth pointmap $\bar{X}$ that supervises the regression loss. It therefore adds no new ground-truth information and requires no additional labels or datasets, it is computed on the fly in the loss. The signal is thus an inductive bias that redistributes existing supervision toward high-curvature regions, rather than a new training signal. We compute K_i on the same `avg_dis`-normalized pointmap as ℓ_i, set border pixels to zero, and restrict to valid pixels.

Outlier Clamp. Discrete curvature is numerically large at depth discontinuities (object silhouettes, occlusion boundaries), where a handful of pixels would otherwise dominate the objective. We therefore clamp the normalized signal at a fixed constant $\tau = 10$, capping the maximum upweighting at any single pixel at $1 + \gamma\tau$. Relaxing regularization near depth edges in this way is a standard robustness measure [20], and we treat τ as a fixed constant rather than a tuned hyperparameter.

Gradient Control. To test whether any improvement is specific to curvature rather than to generic edge emphasis, we evaluate an identical objective with a first-order signal, $S_i = \|\nabla \bar{X}_i\|$ (the pointmap gradient magnitude), in place of the second-order curvature $|K_i|$.

4.3 Implementation

The curvature signal is implemented as a small addition to the D^{2}USt3R loss module, without touching the network, the data loader, or the training loop. Given the ground-truth pointmap tensor $\bar{X} \in \mathbb{R}^{B\times 3\times H\times W}$ that the existing loss receives, the four-neighbor umbrella Laplacian is one shifted subtraction. Below, $\odot$ denotes the elementwise (Hadamard) product, defined for two tensors of equal shape by $(A \odot B)_i = A_i B_i$.

Algorithm 1 Curvature-weighted loss, drop-in replacement for the position term in D^{2}USt3R's `loss.py`.

1: **Input:** predicted $\hat{X}$, ground truth $\bar{X}$ (both `avg_dis`-normalized), valid mask M, confidence c, hyperparameters γ, τ, α
2: $\ell \leftarrow \|\hat{X} - \bar{X}\|_2$ ▷ position residual
3: $\bar{X}^{\uparrow}, \bar{X}^{\downarrow}, \bar{X}^{\leftarrow}, \bar{X}^{\rightarrow} \leftarrow \mathrm{roll}(\bar{X})$ ▷ four shifts, one line each
4: $K \leftarrow \frac{1}{2}\left\|\frac{1}{4}(\bar{X}^{\uparrow}+\bar{X}^{\downarrow}+\bar{X}^{\leftarrow}+\bar{X}^{\rightarrow}) - \bar{X}\right\|$
5: $K \leftarrow K \odot M_{\mathrm{border}}$ ▷ zero the one-pixel border
6: $\bar{K} \leftarrow \mathrm{mean}(K \odot M)$ ▷ mean over valid pixels
7: $w \leftarrow 1 + \gamma \cdot \min(K/\bar{K},\ \tau)$
8: **return** $\mathrm{mean}\big[\, c \cdot (w \odot \ell) - \alpha \log c \,\big]_M$

The computation is a fixed number of tensor operations at the training resolution 512×288 and adds well under one percent to the wall-clock time of a training step, dominated as it is by the forward and backward pass of the transformer decoder and DPT heads. No caching, precomputation, or additional data loading is needed, and the signal is recomputed each step directly from the ground-truth pointmap the loss receives. Setting $\gamma = 0$ leaves $w \equiv 1$ and reduces line 8 to the standard DUSt3R confidence-weighted regression.

5 Experiments

5.1 Training Setup

PointOdyssey [56] is a large-scale synthetic dataset of dynamic scenes with dense ground-truth depth and camera poses. TartanAir [52] is a large-scale synthetic dataset of predominantly static scenes built for visual SLAM. We use these two because they are two of the five synthetic datasets D^2USt3R's own training recipe draws on [19], so our finetune narrows that training distribution rather than introducing a new one. We finetune from the released D^2USt3R checkpoint with the encoder frozen and the decoder and DPT heads trainable, training on both at resolution 512×288 (6,000 PointOdyssey and 2,000 TartanAir two-view clips per epoch, temporal strides $\{2, 4, 6, 8\}$ for PointOdyssey and 8 for TartanAir), batch size 2 with gradient accumulation of 8 (effective batch 16), learning rate 3×10^{-5} with a cosine schedule to 1×10^{-6} and one warmup epoch, mixed precision, for 12 epochs, on a single 20 GB GPU. We run the matched $\gamma = 0$ baseline, the $\gamma = 1$ curvature arm, and the first-order gradient control over five random seeds each, and a two-seed sweep over $\gamma \in \{0.5, 2\}$.

5.2 Hardware and Runtime

All runs use a single NVIDIA RTX 4000 Ada Generation GPU (20 GB). Because the encoder is frozen and only the decoder and DPT heads are trained, each run fits within memory at batch size 2 with gradient accumulation of 8. The study comprises 19 finetuning runs (the $\gamma = 0$, $\gamma = 1$, and gradient-control arms at five seeds each, and the $\gamma \in \{0.5, 2\}$ sweep at two seeds). A single 12-epoch run takes about 7.4 hours, for roughly 164 GPU-hours in total.

5.3 Evaluation Protocol

For each evaluation clip we take the predicted depth of the first view (the z-coordinate of its predicted pointmap) and align it to ground-truth by per-frame least-squares scale-and-shift alignment, the standard scale-shift-invariant protocol [39]. We discard pixels whose aligned prediction is non-positive and, on Sintel, cap ground-truth depth at 70 m so that a few far-depth pixels do not dominate the mean error. We report two metrics over valid pixels. Absolute relative error (lower is better) is

$$\mathrm{AbsRel} = \frac{1}{N} \sum_i \frac{|\hat{d}_i - d_i|}{d_i}, \tag{6}$$

and the inlier ratio (higher is better) is

$$\delta_1 = \frac{1}{N} \sum_i \mathbf{1}\big[\max(\hat{d}_i/d_i,\, d_i/\hat{d}_i) < 1.25\big]. \tag{7}$$

Each metric is reported over all valid pixels and, separately, over a boundary subset formed by the top 10% of valid pixels by ground-truth curvature magnitude $|K_i|$, the region where the inductive bias is expected to act.

We evaluate under two regimes. PointOdyssey-val consists of held-out PointOdyssey sequences that are excluded from training but drawn from the training distribution, so it measures in-distribution accuracy. Sintel [5], a synthetic benchmark of dynamic scenes with dense ground-truth depth, is not seen during training and measures zero-shot generalization to a new distribution.

Table 1: Matched finetune on PointOdyssey and TartanAir, evaluated with scale-and-shift alignment. All arms are mean$_{\pm\text{std}}$ over five seeds. Bold marks the best of the three finetune arms in each column. The base checkpoint is a reference row and is not bolded. Curvature weighting ($\gamma = 1$) lowers error over the $\gamma = 0$ baseline on seven of the eight metrics. The exception is in-distribution δ_1 over all pixels. The gains are largest and most consistent zero-shot on Sintel (5.7% lower AbsRel over all pixels, 6.0% at boundaries), and smaller in distribution (0.8% over all pixels, 2.4% at boundaries). Curvature takes the best value in six of the eight columns, and the baseline and gradient control in one each. Curvature-versus-gradient differences fall within the seed standard deviation. †The base checkpoint was trained on more data than our finetune, so it serves as a reference only. The controlled comparison is $\gamma = 0$ versus $\gamma = 1$. These arms share training data.

	PointOdyssey-val (in-distribution)				Sintel (zero-shot)			
	All		Boundary		All		Boundary	
Method	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
Base checkpoint†	0.0770	0.9367	0.0709	0.9464	0.3243	0.6744	0.2202	0.6790
$\gamma = 0$ (baseline)	$0.0826_{\pm.003}$	$\mathbf{0.9315}_{\pm.004}$	$0.0714_{\pm.002}$	$0.9460_{\pm.002}$	$0.4939_{\pm.022}$	$0.5754_{\pm.010}$	$0.2962_{\pm.009}$	$0.5881_{\pm.019}$
$\gamma = 1$ (curvature)	$\mathbf{0.0819}_{\pm.003}$	$0.9286_{\pm.005}$	$\mathbf{0.0697}_{\pm.002}$	$0.9466_{\pm.003}$	$\mathbf{0.4658}_{\pm.018}$	$\mathbf{0.6027}_{\pm.015}$	$\mathbf{0.2783}_{\pm.008}$	$\mathbf{0.6199}_{\pm.021}$
∇ gradient control	$0.0867_{\pm.007}$	$0.9293_{\pm.005}$	$0.0700_{\pm.002}$	$\mathbf{0.9478}_{\pm.001}$	$0.4774_{\pm.021}$	$0.5968_{\pm.019}$	$0.2829_{\pm.012}$	$0.6191_{\pm.026}$

Table 2: Weight-strength sweep, all-pixel metrics. The $\gamma = 0$ and $\gamma = 1$ rows repeat the five-seed means from Table 1. The $\gamma = 0.5$ and $\gamma = 2$ rows are two-seed means and show a trend, not a matched comparison. In-distribution AbsRel stays flat across γ, within 0.001. Zero-shot AbsRel decreases monotonically and reaches 0.425 at $\gamma = 2$.

	PointOdyssey-val		Sintel	
γ	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
0	0.0826	**0.9315**	0.4939	0.5754
0.5	0.0827	0.9312	0.4755	0.6061
1	**0.0819**	0.9286	0.4658	0.6027
2	0.0825	0.9303	**0.4246**	**0.6277**

5.4 Findings

Three results follow from Tables 1 and 2. First, weighting the position loss by local geometric magnitude helps. The $\gamma = 1$ arm beats the matched $\gamma = 0$ baseline on seven of the eight metrics, the exception being in-distribution δ_1 over all pixels. The gains are largest and most consistent in the zero-shot regime, where the AbsRel and δ_1 improvements on Sintel exceed one standard deviation across the five seeds; the in-distribution improvements fall within seed noise. All finetunes stay below the base checkpoint on zero-shot Sintel, an effect of finetuning on a narrow domain, so we read the matched $\gamma = 0$ vs. $\gamma = 1$ delta rather than the comparison to the base checkpoint.

Second, curvature is the strongest of the three signals. The $\gamma = 1$ curvature arm attains the best value in six of the eight columns, including all four zero-shot columns, while the $\gamma = 0$ baseline and the first-order gradient control each lead in a single in-distribution column. The gradient control weights by the pointmap gradient magnitude $\|\nabla \bar{X}\|$ with everything else fixed, and is the closest competitor, with curvature-versus-gradient differences falling within the seed standard deviation (Table 1). The effect operates through the redistribution of supervision toward geometrically complex regions, which a first-order signal also captures, so we read the second-order form as the better-performing choice rather than a necessary one.

Third, the weight strength does not trade off in-distribution accuracy (Table 2). In-distribution AbsRel is flat across γ, within 0.001, while zero-shot AbsRel falls monotonically through $\gamma = 2$.

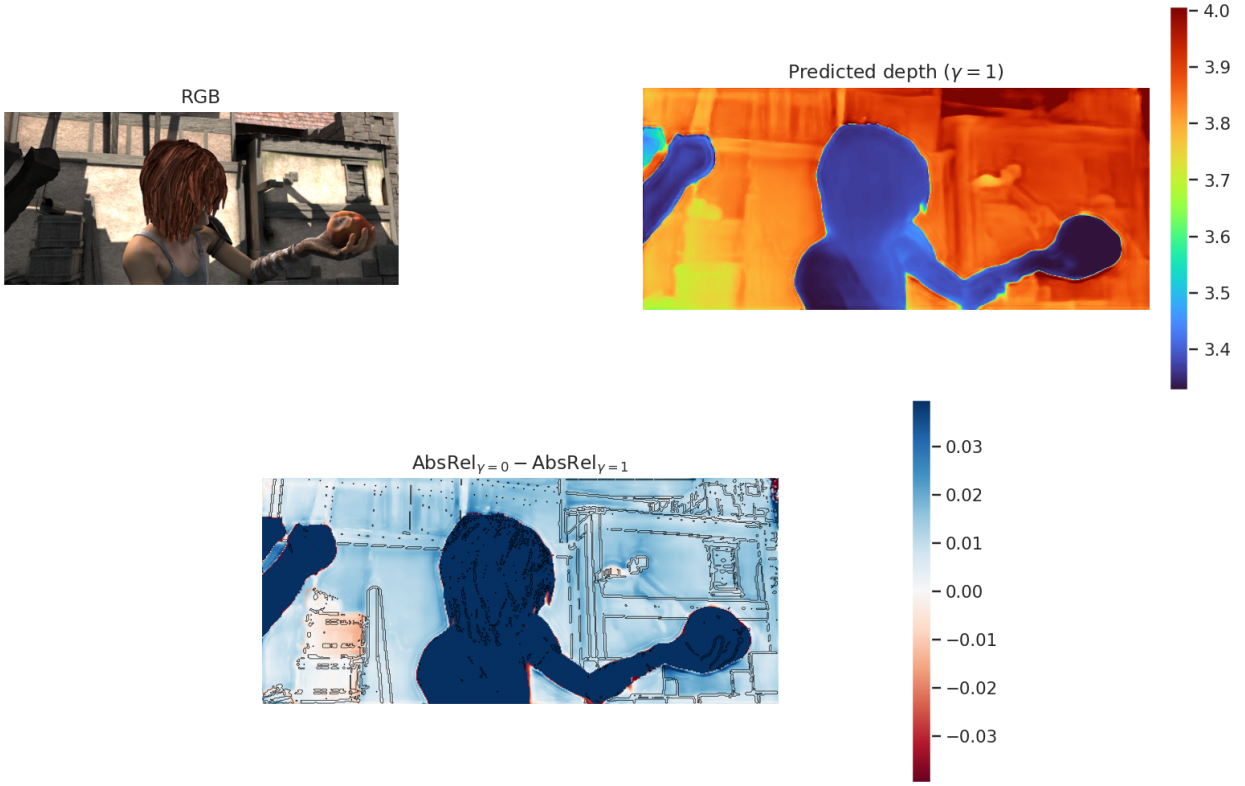


Figure 2: Where curvature weighting helps, on a zero-shot Sintel frame (`alley_1`). From left: the input image, the predicted depth of the curvature arm ($\gamma = 1$), and the per-pixel AbsRel difference $\mathrm{AbsRel}_{\gamma=0} - \mathrm{AbsRel}_{\gamma=1}$ under scale-and-shift alignment. The difference is positive, and blue, wherever the curvature arm has the lower error. The black outline marks the top-decile high-curvature boundary set where reduction is concentrated.

Reading the boundary and all-pixel columns of Table 1 together localizes the gain. In-distribution, curvature weighting reduces AbsRel by 0.8% over all pixels and by 2.4% on the boundary subset; zero-shot, by 5.7% over all pixels and 6.0% at boundaries. The boundary subset is the top decile of pixels by $|K_i|$ and the weight w_i grows with $|K_i|$, so a gain concentrated there is the signature the mechanism predicts (Fig. 2). The in-distribution ratio (2.4% against 0.8%) shows this concentration, although the in-distribution gains themselves fall within seed noise; the gradient control shows the same pattern. Both signals therefore act through one channel, the redistribution of supervision toward geometrically complex regions, rather than through anything specific to second-order geometry. The three results give a controlled account of the curvature-weighted loss: a positive effect that is clearest in the zero-shot regime and concentrated at the pixels the weighting emphasizes, a monotonic zero-shot trend, and a mechanism attributable to geometric reweighting.

6 Conclusion

Reweighting the pointmap position loss by the local mean curvature of the ground-truth surface yields a small but consistent depth improvement, obtained at no cost in parameters, data, or annotation. Against the matched baseline the curvature arm lowers error on seven of the eight metrics we report, and curvature attains the best value of the three arms on six of the eight. The effect concentrates in the zero-shot regime, where the improvement on Sintel exceeds the five-seed standard deviation, while

the in-distribution gains are present but remain within that noise. Zero-shot error also decreases monotonically across the weight-strength sweep. A first-order gradient control recovers much of the improvement, so the mechanism is the redistribution of supervision toward geometrically complex regions rather than a property of second-order geometry alone. Curvature is the stronger of the two signals and the one we recommend, though it is not the only reweighting that helps.

Two limitations bound these conclusions. First, the baseline is the released D^2USt3R checkpoint finetuned with the standard confidence-weighted regression, not a reimplementation of the static and dynamic alignment objective of D^2USt3R, whose training code was not released. The study therefore measures the effect of curvature weighting on that regression and does not claim an improvement over D^2USt3R itself. Second, because a first-order gradient signal reproduces most of the gain, the benefit is best read as a general property of geometry-aware reweighting rather than as evidence for second-order curvature in particular.

Several directions remain open. First, curvature enters here only as a weight on the position loss. Predicting it explicitly as an auxiliary output, with its own head alongside the existing dynamic-mask head of D^2USt3R, would turn the ground-truth curvature signal into a target rather than a weight. Second, mean curvature is a property of the surface rather than of its pose, but this invariance holds only under rigid motion. A deforming surface changes shape between frames, so a cross-frame curvature-consistency term would first have to separate genuine deformation from prediction error before it could supervise the dynamic regions in which the rigid ground-truth construction of Eq. (1) is invalid. Third, the signal touches only the pairwise training loss. Carrying it into the global-alignment stage, in which pairwise pointmaps are fused across a whole scene, would let curvature shape the multi-view reconstruction directly at the cost of a higher-order consistency term than the pairwise one used here.

References

[1] Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9535–9545, 2024. 2, 4

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003. 4

[3] Yizhak Ben-Shabat and Stephen Gould. Deepfit: 3d surface fitting via neural network weighted least squares. In *European conference on computer vision*, pages 20–34. Springer, 2020. 2, 4

[4] Kristian Bredies, Karl Kunisch, and Thomas Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010. 4

[5] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 8

[6] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006. 4

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016. 4

[8] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, 1999. 4

[9] Thanuja Dharmasiri, Andrew Spek, and Tom Drummond. Joint prediction of depths, normals and surface curvature from rgb images using cnns. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1505–1512. IEEE, 2017. 4

[10] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 4

[11] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 4

[12] David F Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3392–3399, 2013. 4

[13] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 3

[14] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017. 4

[15] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019. 4

[16] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 4

[17] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5511–5520, 2022. 4

[18] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and computational harmonic analysis*, 30(2):129–150, 2011. 4

[19] Jisang Han, Honggyu An, Jaewoo Jung, Takuya Narihira, Junyoung Seo, Kazumi Fukuda, Chaehyun Kim, Sunghwan Hong, Yuki Mitsufuji, and Seungryong Kim. Enhancing 3d reconstruction for dynamic scenes. *Advances in Neural Information Processing Systems*, 38:1210–1234, 2026. 2, 3, 5, 8

[20] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2360–2367, 2013. 4, 7

[21] Mark Kac. Can one hear the shape of a drum? *The american mathematical monthly*, 73(4P2):1–23, 1966. 4

[22] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 4

[23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3

[24] Georg Kuschk and Daniel Cremers. Fast and accurate large-scale stereo reconstruction using variational methods. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 700–707, 2013. 4

[25] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024. 3

[26] Bruno Lévy. Laplace-beltrami eigenfunctions towards an algorithm that" understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 13–13. IEEE, 2006. 4

[27] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1119–1127, 2015. 4

[28] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8456–8465, 2023. 4

[29] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Wei Li, Christian Theobalt, Ruigang Yang, and Wenping Wang. Adaptive surface normal constraint for depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12849–12858, 2021. 4

[30] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003. 2, 4, 5, 6

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[32] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389, 2006. 4

[33] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 4

[34] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5865–5874, 2021. 4

[35] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10318–10327, 2021. 4

[36] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 283–291, 2018. 2, 4

[37] Michael Ramamonjisoa and Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2, 4

[38] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 4

[39] 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. 4, 8

[40] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. Laplace–beltrami spectra as 'shape-dna'of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006. 4

[41] Szymon Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.*, pages 486–493. IEEE, 2004. 4

[42] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 3

[43] Duochao Shi, Weijie Wang, Donny Y Chen, Zeyu Zhang, Jia-Wang Bian, and Bohan Zhuang. Revisiting depth representations for feed-forward 3d gaussian splatting. In *2026 International Conference on 3D Vision (3DV)*, pages 1071–1080. IEEE, 2026. 4

[44] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004. 4

[45] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009. 4

[46] Marshall F Tappen. Recovering shape from a single image of a mirrored surface from curvature constraints. In *CVPR 2011*, pages 2545–2552. IEEE, 2011. 2, 4

[47] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995. 4

[48] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(11):9426–9437, 2024. 4

[49] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 3

[50] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 4

[51] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024. 2, 3, 5

[52] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 8

[53] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 3

[54] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022. 2, 4

[55] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Forrester Cole, Deqing Sun, Ming-Hsuan Yang, et al. Monst3r: A simple approach for estimating geometry in the presence

of motion. In *International Conference on Learning Representations*, volume 2025, pages 82863–82886, 2025. 2, 3, 5

[56] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. 8