CLICK3R: Interactive Stereo 3D Reconstruction with Sparse Correspondence Clicks

Hao Gu[*†] **Jin Yao**[*] **Zezhou Cheng**
University of Virginia
https://uva-computer-vision-lab.github.io/click3r

Abstract

Despite impressive progress, existing feed-forward large 3D reconstruction models produce a single, irrevocable prediction. When these models fail, there is no mechanism to correct their mistakes. To address this limitation, we introduce CLICK3R, a framework for *interactive geometric correction* of feed-forward models using sparse human correspondence clicks. Given an image pair and a small set of cross-view point correspondences provided by a user, CLICK3R injects geometric constraints into the 3D reconstruction networks through a lightweight point correspondence adapter, enabling the model to resolve ambiguities in challenging scenarios such as repetitive textures, symmetric structures, large viewpoint changes, or textureless regions. To evaluate interactive reconstruction, we curate a challenging benchmark specifically designed to expose failure modes of existing feed-forward 3D reconstruction methods. Experiments show that CLICK3R reduces reconstruction error with as few as a single correspondence click, improving performance on both standard and our curated benchmarks. Our results highlight that sparse human interaction provides an effective and practical mechanism for correcting geometric reconstruction errors.

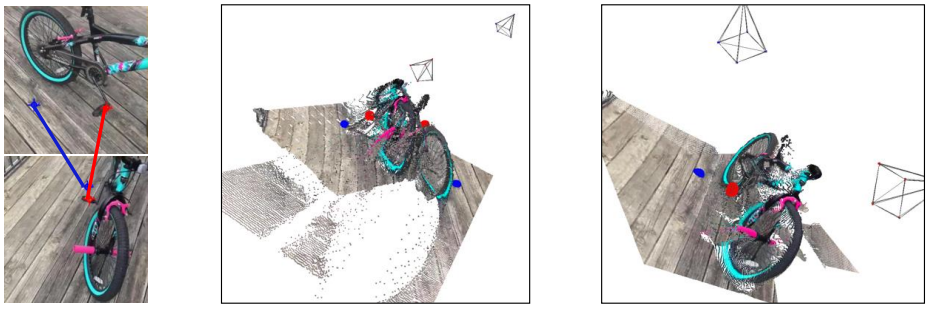
1 Introduction

Feed-forward stereo 3D reconstruction methods [44, 16, 29, 49] have transformed multi-view geometry by replacing iterative optimization [25, 34] with direct regression. Given two images, these models predict dense 3D pointmaps in a single forward pass, achieving remarkable speed and generalization across diverse scenes. This paradigm has recently been extended to multi-view settings [42, 45, 53, 40, 3]. Despite their impressive performance, the efficiency comes at a fundamental cost: *when the prediction is wrong, there is no simple mechanism to correct it.* Unlike optimization-based methods [34] that iteratively refine their estimates using geometric constraints, feed-forward models produce a single, irrevocable output and fail silently, particularly on challenging scenes with repetitive textures, symmetric structures, or extreme viewpoint changes (Fig. 1a,b).

Several approaches aim to improve the reliability of feed-forward reconstruction models through test-time optimization [24, 5]. For instance, LoRA3D [24] introduces a self-calibration pipeline that fine-tunes a pretrained DUSt3R model [44] to a target scene. Although these methods improve performance, iterative test-time optimization is computationally expensive and, more importantly, cannot resolve failures when the model encounters genuinely ambiguous scenes at inference time. Another line of work improves robustness by incorporating auxiliary geometric priors. For example, Pow3R [10] shows that conditioning on camera intrinsics, relative camera poses, or depth maps can significantly improve reconstruction performance. However, such auxiliary information is often unavailable in practice. Meanwhile, classical reconstruction methods have long relied on explicit

[*]Equal contribution. [†]Project lead.

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



(a) Input Views w/ Correspondence Clicks

(b) Prior State-of-the-Art **without** Correspondence Clicks

(c) Click3R: Interactive 3D Reconstruction **with** Correspondence Clicks

Figure 1: **Overview.** Given (a) an image pair and sparse correspondences (*colored dot clicks*), CLICK3R conditions DUSt3R [44] to produce accurate 3D reconstruction. (b) DUSt3R fails on a scene with limited overlaps. (c) With as few as two clicks, CLICK3R recovers correct geometry.

correspondences as geometric constraints [34, 31, 22]. Unfortunately, automatically estimated correspondences remain unreliable in scenarios with repetitive textures or large viewpoint changes.

Nevertheless, cross-view correspondences, even just a handful of point pairs, carry powerful geometric information: they constrain the epipolar geometry, resolve depth ambiguities, and disambiguate symmetric structures. This motivates a simple question: **can sparse human-provided correspondences guide feed-forward 3D reconstruction models without sacrificing the efficiency of the feed-forward paradigm?** Instead of relying solely on feed-forward models or automatically estimated correspondences, we allow the user to manually provide a small set of cross-view point correspondences (Fig. 1a), when the model produces unsatisfactory reconstructions.

To unlock this capability, we introduce **CLICK3R**, a lightweight framework that incorporates user-provided correspondences into feed-forward 3D reconstruction, enabling **interactive geometric correction without test-time model fine-tuning or optimization**. Specifically, CLICK3R employs a simple correspondence adapter that injects the geometric constraints imposed by user clicks into the reconstruction network via a cross-attention mechanism (Fig. 2). We train the base model and the adapter jointly end-to-end on widely used 3D datasets [28, 2, 57] using simulated human clicks. During training, the number of clicks is randomly sampled to expose the model to varying levels of user interaction, enabling it to operate under different levels of click sparsity at test time.

To systematically evaluate the proposed interactive reconstruction, we further introduce **Hard3DBench**, a benchmark designed to expose failure modes of state-of-the-art feed-forward reconstruction models. The benchmark contains image pairs and 5-view tuples with ground-truth 3D reconstructions, exhibiting challenging conditions (large viewpoint changes, low overlap, large baselines, extreme depth ranges) where feed-forward reconstruction models often struggle. Our experiments demonstrate that CLICK3R consistently outperforms baseline methods, particularly on challenging scenarios from Hard3DBench (Fig. 1c).

Our contributions are summarized as follows:

- We propose CLICK3R, which injects sparse human correspondence clicks into feed-forward stereo 3D reconstruction networks via a lightweight correspondence adapter.
- We curate Hard3DBench, a diagnostic benchmark with a detailed failure taxonomy under which current feed-forward reconstruction models degrade substantially relative to standard evaluation samples.
- We show that CLICK3R dramatically improves reconstruction accuracy on challenging cases. To the best of our knowledge, this is the first work on interactive stereo feed-forward 3D reconstruction using sparse correspondence clicks.

2 Related Work

Feed-Forward Stereo 3D Reconstruction. DUSt3R [44] pioneered dense 3D pointmap regression from image pairs using a CroCo-pretrained [48, 47] ViT encoder-decoder, bypassing explicit matching and triangulation. MASt3R [16] extended this with dense local feature heads. The paradigm has rapidly advanced toward multi-view: Spann3R [41] and MUSt3R [3] use spatial memory across frames; CUT3R [43] employs a recurrent model with persistent state; Fast3R [53] scales to 1000+ images in a single forward pass; and VGGT [42] and Pi3 [45] predict from arbitrary views with a single Transformer. Despite their diversity, all these methods rely solely on learned visual priors and cannot incorporate external geometric evidence at inference. Our work is orthogonal: CLICK3R augments feed-forward stereo reconstruction with sparse correspondence guidance.

Conditioned 3D Reconstruction. Several concurrent methods condition feed-forward 3D models on external geometric signals: Pow3R [10] on camera intrinsics, depth, or pose, G-CUT3R [14] on camera and depth modalities, Rig3R [17] on multi-camera rig metadata, and Mono3R [18] on monocular depth. MapAnything [13] unifies a flexible mix of conditioning inputs (intrinsics, poses, depth, partial reconstructions) into a single feed-forward model, while Any4D [12] extends conditional reconstruction to dynamic 4D scenes with multi-modal sensor inputs. In the depth estimation domain, PromptDA [21] prompts Depth Anything [54] with sparse LiDAR points. All these methods condition on *scene-level* priors that improve metric accuracy but cannot resolve perceptual aliasing. To our knowledge, CLICK3R is the first to introduce *correspondence prompting*, which conditions feed-forward 3D reconstruction on cross-view point correspondences that directly disambiguate geometric ambiguities, analogous to SAM's [15, 27] click-based segmentation prompting.

Interactive Vision System. Interactive segmentation has a long history of incorporating user guidance into visual recognition. Early methods use bounding boxes [30] or scribbles [20], while recent approaches adopt click-based prompting [51, 4, 23, 39], culminating in SAM [15, 27]. Beyond segmentation, point-based interaction has been extended to image editing: DragGAN [26] and DragDiffusion [37] enable users to manipulate content by dragging points. Recent work has also explored interactive guidance for 3D reconstruction; for example, Toon3D [46] uses sparse user annotations for cartoon 3D reconstruction. However, these interactive paradigms have largely remained confined to 2D tasks or stylized 3D settings. Early interactive systems also explored correspondence-based 3D reconstruction [9]. CLICK3R brings this idea to feed-forward stereo models, with point correspondences as geometric constraints rather than semantic cues.

3 Methodology

Given an image pair (I_1, I_2) and an optional set of sparse point correspondences $\mathcal{C} = \{(\mathbf{p}_i^1, \mathbf{p}_i^2)\}_{i=1}^K$, where $\mathbf{p}_i^v \in \mathbb{R}^2$ denotes the pixel coordinates in view v, CLICK3R predicts dense 3D pointmaps for both views. When no correspondences are provided, the adapter reduces to an identity mapping, and the model behaves identically to the base reconstruction model without prompts. An overview of the CLICK3R architecture is shown in Fig. 2.

3.1 Preliminaries: DUSt3R

DUSt3R [44] uses a shared ViT-Large encoder ($D_{\text{enc}} = 1024$, 24 blocks) to process each image independently, followed by an asymmetric decoder ($D_{\text{dec}} = 768$, 12 blocks per branch) with cross-attention between views. The encoder produces patch tokens

$$\mathbf{F}_v = \text{Encoder}(I_v) \in \mathbb{R}^{N \times D_{\text{enc}}}, \quad N = \tfrac{H}{p} \times \tfrac{W}{p}, \tag{1}$$

where p is the patch size. Tokens are projected to decoder dimension, $\mathbf{G}_v = \text{proj}(\mathbf{F}_v) \in \mathbb{R}^{N \times D_{\text{dec}}}$, then processed by interleaved self- and cross-attention blocks. A linear prediction head maps the decoded tokens to dense 3D pointmaps $\hat{X}_v \in \mathbb{R}^{H \times W \times 3}$ and confidence maps $\hat{C}_v \in \mathbb{R}^{H \times W}$ in view 1's frame.

CLICK3R inserts its correspondence adapter after this projection and before the first decoder block, modifying $\mathbf{G}_v$ using sparse correspondence prompts. The base model and adapter are trained jointly end-to-end.

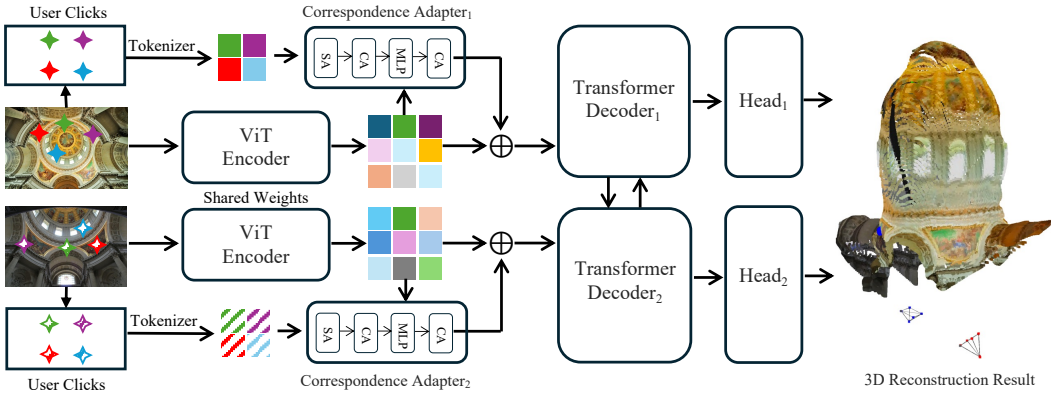


Figure 2: **CLICK3R architecture.** A shared ViT encoder processes the two input images. Correspondence prompts are tokenized using our correspondence tokenizer (Section 3.2). A symmetric point correspondence adapter, implemented as a lightweight transformer with self-attention and cross-attention (Section 3.3), updates the decoder-input features for each view. The asymmetric decoder then interprets the updated embeddings and passes them to the original DUSt3R [44] heads to predict 3D pointmaps and confidence.

3.2 Correspondence Tokenizer

Our tokenizer converts sparse correspondences into tokens for cross-attention with image features. For each correspondence point $\mathbf{p}_i^v = (x_i, y_i)$, we normalize its coordinates to the token grid:

$$\bar{\mathbf{p}}_i^v = \left(\frac{x_i/p_w}{W_t - 1}, \frac{y_i/p_h}{H_t - 1} \right), \tag{2}$$

clipped to $[0, 1]$, where p_w, p_h are effective patch sizes and W_t, H_t are token-grid dimensions; this is the scale on which the image positional encoding places token j at $j/(W_t - 1)$. We encode the normalized point using SAM-style random Fourier features [15]:

$$\mathbf{e}_{\text{pos}}(\bar{\mathbf{p}}_i^v) = [\sin(2\pi \mathbf{B}(2\bar{\mathbf{p}}_i^v - 1)); \cos(2\pi \mathbf{B}(2\bar{\mathbf{p}}_i^v - 1))] \in \mathbb{R}^D, \tag{3}$$

with fixed Gaussian matrix $\mathbf{B} \in \mathbb{R}^{D/2 \times 2}$, preserving sub-pixel precision.

To identify the view and to link matched points across views, we add a learnable image embedding $\mathbf{e}_{\text{img}}(v) \in \mathbb{R}^D$ and a learnable pair embedding $\mathbf{e}_{\text{pair}}(i) \in \mathbb{R}^D$ shared by both points in the same correspondence. The final token is

$$\mathbf{t}_i^v = \mathbf{e}_{\text{pos}}(\bar{\mathbf{p}}_i^v) + \mathbf{e}_{\text{img}}(v) + \mathbf{e}_{\text{pair}}(i) \in \mathbb{R}^D. \tag{4}$$

For each view, visible correspondence tokens form $\mathbf{T}_v \in \mathbb{R}^{K_v \times D}$. The tokenizer is shared across both view adapters.

3.3 Point Correspondence Adapter

We inject correspondence information into DUSt3R's decoder using a two-way attention adapter inspired by SAM [15], enabling bidirectional exchange between sparse correspondence tokens and dense image tokens (Figure 2).

Two-Way Attention. The adapter has $L = 2$ layers. Each layer applies: (1) self-attention over prompt tokens $\mathbf{T}_v$; (2) prompt-to-image cross-attention, $\text{Attn}(\mathbf{T}_v, \mathbf{G}_v, \mathbf{G}_v)$; (3) an MLP on prompt tokens; and (4) image-to-prompt cross-attention, $\text{Attn}(\mathbf{G}_v, \mathbf{T}_v, \mathbf{T}_v)$. This bidirectional design lets prompts condition image features with correspondence constraints while allowing prompt tokens to absorb local visual context.

Image Positional Encoding. For cross-attention, image positional encodings are computed by applying the same Fourier PE to normalized DUSt3R token-grid positions. Thus, prompt and image tokens share a common coordinate space, allowing cross-attention to compare their spatial locations consistently.

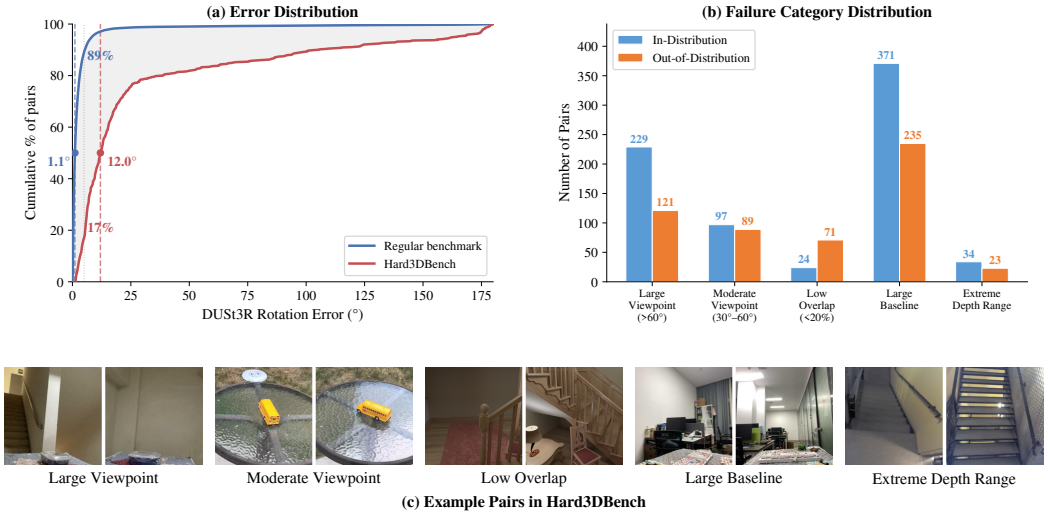


Figure 3: **Hard3DBench.** Statistics shown for the 2-view stereo subset of Hard3DBench. (a) Hard3DBench produces substantially larger pretrained DUSt3R rotation errors than the regular benchmark. (b) Failure category distribution. (c) Example image pairs in Hard3DBench.

Asymmetric Design. We employ independent adapters $\mathcal{A}_1, \mathcal{A}_2$ for view 1 and view 2, each with a separate two-way attention adapter but sharing the same correspondence tokenizer:

$$\mathbf{G}'_1 = \mathcal{A}_1(\mathbf{G}_1,\ \mathbf{T}_1,\ \text{pos}_1) \tag{5}$$

$$\mathbf{G}'_2 = \mathcal{A}_2(\mathbf{G}_2,\ \mathbf{T}_2,\ \text{pos}_2) \tag{6}$$

This reflects DUSt3R's asymmetric decoder, where each view has a separate branch. When no prompts are provided ($K = 0$, $\mathbf{T}_v = \varnothing$), the adapter reduces to an identity mapping: $\mathbf{G}'_v = \mathbf{G}_v$.

3.4 Training Details

We use DUSt3R's confidence-weighted regression loss:

$$\mathcal{L} = \sum_{v\in\{1,2\}} \sum_{\mathbf{x}} C_v(\mathbf{x}) \cdot \|\hat{X}_v(\mathbf{x}) - X^*_v(\mathbf{x})\|_2 - \alpha \log C_v(\mathbf{x}) \tag{7}$$

where C_v is predicted confidence, $\hat{X}_v$ and X^*_v are predicted and ground-truth pointmaps, and $\alpha = 0.2$. The loss is computed in a scale invariant manner.

We generate training prompts from ground-truth 3D geometry by sampling a curriculum over the number of correspondences and keeping only mutually visible correspondences. See more training and prompt generation details in B.

4 Hard3DBench

The primary value of CLICK3R is on challenging scenarios where DUSt3R fails. Standard evaluations report only aggregate metrics, masking catastrophic failures on geometrically ambiguous scenes. To enable systematic analysis, we introduce **Hard3DBench**, a curated benchmark for both *2-view stereo* and *5-view multi-view* settings, accompanied by a *failure taxonomy* that exposes the specific failure modes of feed-forward 3D reconstruction. The 2-view setting contains 404 in-distribution and 300 out-of-distribution pairs, and the 5-view setting 400 in-distribution and 300 out-of-distribution tuples. In-distribution samples come from held-out test/validation splits of four datasets used to train DUSt3R and CLICK3R: BlendedMVS [56], CO3D [28], Habitat [32], and WildRGB-D [50]. Out-of-distribution samples come from five datasets absent from training: 7Scenes [38], DTU [11], ETH3D [35], NRGBD [1], and ScanNet [6]. Full curation details are in Appendix C.

Failure Taxonomy. We identify five primary failure modes in the 2-view subset (Figure 3): *large viewpoint change* ($> 60°$), *moderate viewpoint change* ($30°$–$60°$), *low overlap* ($< 20\%$), *large*

Table 1: **Standard benchmark results** on held-out splits of CO3D, ScanNet++, and ARKitScenes. All methods use 224×224 inputs; gray rows are VGGT and Pi3X at both 518 (native) and 224 resolutions, shown only as reference and not included in the bold/underline ranking. DUSt3R (retrained) is our re-trained DUSt3R baseline using the same training data as CLICK3R for fair comparison; $\pm$x.x denotes the mAA@30 difference from this baseline. PM Err.: median pointmap error. Best in **bold**, second best underlined.

	CO3D		ScanNet++		ARKitScenes	
Method	mAA@30°↑	PM Err.↓	mAA@30°↑	PM Err.↓	mAA@30°↑	PM Err.↓
VGGT [42] (518)	87.76	0.498	59.47	0.261	77.91	0.159
Pi3X [45] (518)	87.95	0.467	80.77	0.140	82.16	0.146
VGGT [42] (224)	62.53	1.994	24.47	1.325	40.89	0.690
Pi3X [45] (224)	85.02	0.577	72.53	0.196	76.39	0.167
DUSt3R [44]	79.20	0.728	82.57	0.190	86.02	0.146
MASt3R [16]	78.45	0.680	51.79	0.459	65.40	0.272
DUSt3R (retrained)	84.33	0.659	90.81	0.174	82.71	0.155
CLICK3R w/o prompts	84.41+0.1	0.678	89.87−0.9	0.175	84.27+1.6	0.157
CLICK3R (1p_best5)	85.71+1.4	**0.651**	91.75+0.9	**0.167**	85.90+3.2	**0.141**
CLICK3R (grid oracle)	**88.21**+3.9	0.658	**93.94**+3.1	0.169	**92.75**+10.0	0.146

baseline, and *extreme depth range*. These categories capture geometric conditions that make correspondence estimation and pose recovery difficult, such as reduced visual matches, amplified triangulation error, and increased ambiguity. Since each category is computed from ground-truth poses and depth, challenging pairs can be classified automatically without manual annotation.

5 Experiments

5.1 Evaluation Setup

Baselines. We compare against state-of-the-art pairwise 3D reconstruction methods: (1) DUSt3R [44], the original pretrained model; (2) DUSt3R (re-trained), the same architecture re-trained on the same data as CLICK3R to isolate the effect of prompting from additional training data; (3) MASt3R [16], which extends DUSt3R with dense local feature matching heads; (4) VGGT [42], a feed-forward vision foundation model for 3D geometry tasks; and (5) Pi3/Pi3X [45], a permutation-equivariant multi-view reconstruction model. VGGT and Pi3/Pi3X use larger architectures and are trained on much more data, including proprietary internal datasets, with much higher computational budgets. We therefore include them as reference rather than apples-to-apples baselines.

Datasets. For standard evaluation, we use held-out splits of three benchmarks: CO3D [28], ScanNet++ [57], and ARKitScenes [2]. For each dataset, we sample 1,000 image pairs with a fixed random seed and use the same pairs across all methods. We additionally evaluate on the Hard3DBench benchmark (Section 5.3).

Metrics. Following DUSt3R [44], we report two metrics: scale-and-shift-invariant pointmap error for 3D reconstruction quality (lower is better) and mean Average Accuracy at threshold $\theta°$ (mAA@θ, higher is better) for relative camera pose accuracy. Tables report mAA@30°; results at 5° and 15° are in Appendix E.2. Implementation details are in Appendix A.

Prompt selection strategies. We evaluate several prompt selection strategies. (a) **Random oracle** (Kp_best$B$): sample K correspondences B independent times and select the set that minimizes reconstruction loss. If there's no best$B$ mentioned, K correspondences are sampled randomly. (b) **Grid oracle**: partition view 1 into a 16×16 grid of 14-pixel cells ($\sim$256 cells), select one random correspondence per populated cell, run a batched forward pass for each candidate, and keep the single correspondence whose cell yields the greatest pose improvement (reduction in $\max(R_{\text{err}}, T_{\text{err}})$) over the no-prompt baseline. Samples without valid correspondences fall back to the no-prompt result.

Table 2: **Hard3DBench results** on the 2-view (left) and 5-view (right) subsets. In-distribution samples come from held-out splits of datasets used to train DUSt3R and CLICK3R; out-of-distribution samples come from datasets absent from training. CLICK3R K denotes K GT correspondences as prompts (per pairwise forward in the 5-view setting). For 5-view evaluation, all DUSt3R-family methods (including CLICK3R) follow DUSt3R's global alignment protocol over the C(5,2) pairwise predictions. Comparison methods use 224×224 resolution except MASt3R (512); gray rows are VGGT and Pi3 at both 518 (native) and 224 resolutions, shown only as reference and not included in the bold/underline ranking. Methods below the dashed line share identical training data; +x.x denotes mAA improvement over DUSt3R (retrained). Pointmap Error is the median across pairs/tuples. Best in **bold**, second best underlined.

	2-view (pair)		*5-view (tuple)*	
Method	mAA@30° ↑	Pointmap Err. ↓	mAA@30° ↑	Pointmap Err. ↓
In-Distribution (404 pairs / 400 tuples)				
VGGT [42] (518)	63.7	0.256	56.5	1.222
Pi3 [45] (518)	64.5	0.232	60.3	0.829
VGGT [42] (224)	40.5	0.620	33.2	1.536
Pi3 [45] (224)	60.7	0.299	56.9	1.234
DUSt3R [44]	39.2	0.502	21.1	2.251
MASt3R [16] (512)	**62.9**	0.229	48.4	**1.333**
DUSt3R (retrained)	56.1	0.216	39.4	2.131
CLICK3R w/o prompts	55.3	0.244	38.8	2.121
CLICK3R (1 prompt)	57.5+1.3	0.218	45.0+5.6	1.883
CLICK3R (4 prompts)	61.3+5.2	0.196	48.6+9.2	1.584
CLICK3R (8 prompts)	61.8+5.7	**0.186**	**49.3**+9.9	1.525
Out-of-Distribution (300 pairs / 300 tuples)				
VGGT [42] (518)	81.3	0.329	81.1	0.099
Pi3 [45] (518)	80.6	0.318	81.4	0.096
VGGT [42] (224)	47.4	1.766	47.1	0.534
Pi3 [45] (224)	71.1	0.433	73.5	0.143
DUSt3R [44]	45.3	0.751	54.7	0.253
MASt3R [16] (512)	51.6	**0.528**	58.0	**0.225**
DUSt3R (retrained)	52.6	0.700	54.8	0.274
CLICK3R w/o prompts	50.8	0.781	54.7	0.310
CLICK3R (1 prompt)	56.6+4.0	0.689	56.7+1.9	0.268
CLICK3R (4 prompts)	60.4+7.8	0.683	58.6+3.8	0.263
CLICK3R (8 prompts)	**62.4**+9.8	0.686	**60.0**+5.2	0.254

5.2 Evaluation on Standard Benchmark

Table 1 presents results on standard benchmarks. Without prompts, CLICK3R performs on par with DUSt3R (retrained). Adding even a single well-chosen prompt (1p_best5) consistently improves over the unprompted baseline. The grid oracle reveals substantially larger headroom, with a striking $+8.5\%$ mAA@30° on ARKitScenes (from 84.27% to 92.75%). This confirms that *prompt location matters significantly*: the model can leverage even a single correspondence to meaningfully reduce pose error when placed in the right region. Notably, the grid oracle delivers the largest pose gains while pointmap error improves only modestly, indicating that prompts primarily reshape the overall scene configuration rather than uniformly tightening local surface fits.

5.3 Evaluation on Hard3DBench

Table 2 reports results on both the 2-view and 5-view subsets. CLICK3R consistently improves mAA@30° over the retrained DUSt3R baseline across all four conditions, with the largest gain of $+9.9$ on the 5-view in-distribution split, and pointmap error also improving in all four conditions. Notably, these improvements are obtained without increasing input resolution or training data. These results suggest that even sparse correspondence prompts effectively resolve failure modes in highly ambiguous scenes, enabling the model to recover accurate geometry where fully automatic methods struggle. Figure 4 shows qualitative comparisons: a few correspondence clicks placed in geometrically

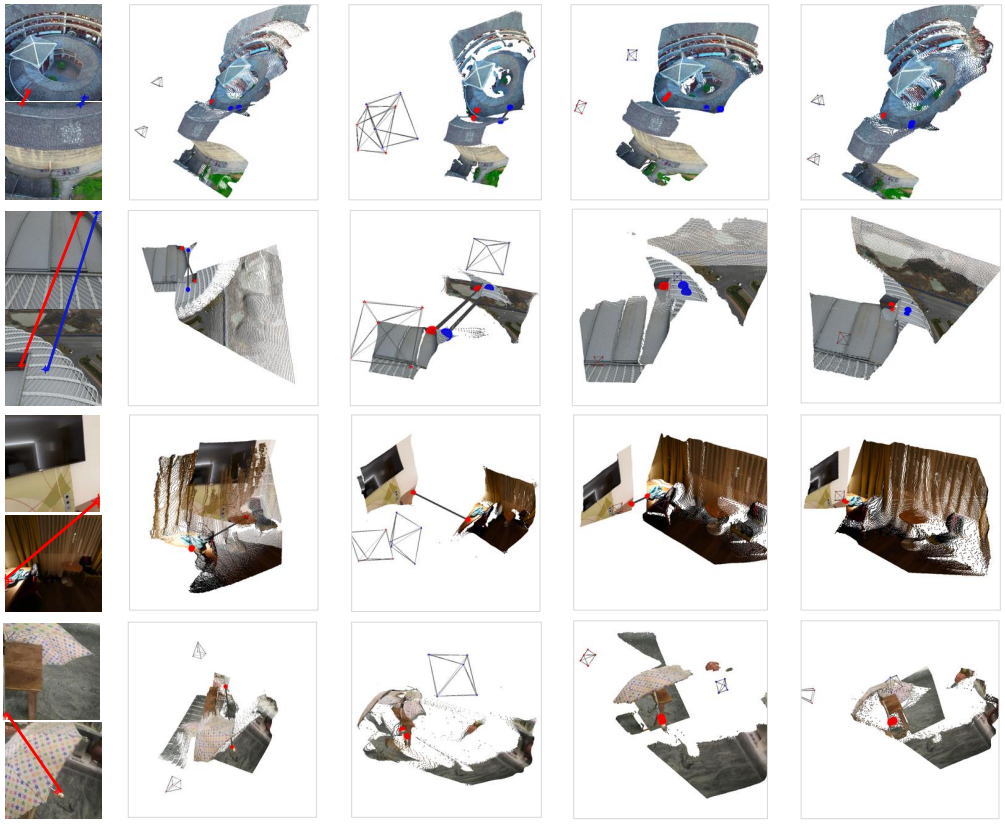


Figure 4: **Qualitative comparison.** Each example features challenging geometry, such as large baselines, limited co-visibility, or substantial viewpoint changes. The input image pair with user-provided correspondence prompts (*colored dots*) is shown on the left, followed by each model's reconstruction. CLICK3R produces more accurate and consistent 3D reconstructions, with prompted correspondences aligning more closely in 3D. Additional qualitative results are provided in Appendix Figure 13.

ambiguous regions resolve failures that produce dramatically incorrect geometry in DUSt3R. See Per-Category Analysis in Appendix Section D.

5.4 Evaluation with Human-Labeled Correspondences

The previous experiments use ground-truth correspondences derived from known 3D geometry. To evaluate CLICK3R under realistic annotation conditions, we test on 100 image pairs with human-labeled correspondences. The pairs are drawn from out-of-distribution set of Hard3DBench and cover challenging geometric configurations. A human annotator, not specifically familiar with the evaluated datasets or scenes, freely chooses sparse correspondences (3.8 per pair on average, from 1 to 7) via a labeling interface (Figure 12), without access to ground-truth depth or camera parameters.

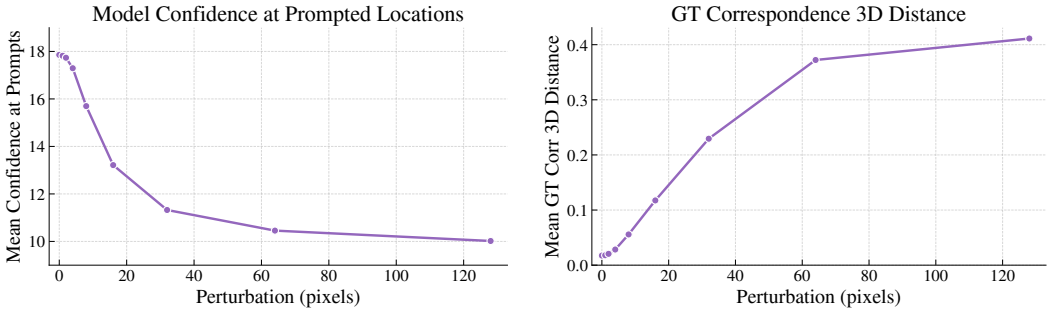
Table 3 demonstrates that human-provided prompts drastically reduce median rotation error compared to the unprompted model (8.78° vs. 17.30°) and raise mAA@30 from 36.2 to 56.2, along with significant improvement in Relative Rotation Accuracy. Compared to the random ground-truth oracle ($K = 4$), human labels recover about 96% of the medRRE improvement gap, showing that sparse, imperfect human clicks capture most of the benefit of perfect correspondences. These results confirm that CLICK3R translates effectively from synthetic oracle prompts to practical human interaction.

5.5 Ablation Study

Training strategy ablation. We compare three training strategies for CLICK3R: (i) freezing the DUSt3R backbone and training only the adapter, (ii) finetuning DUSt3R jointly with the adapter, and (iii) training the full pipeline from CroCo v2 [47] initialization. Averaged over the three standard

Table 3: **Human-labeled correspondence evaluation.** 100 out-of-distribution pairs with human-annotated correspondences. CLICK3R (human) uses sparse clicks from a human annotator; CLICK3R (random GT) uses $K=4$ randomly sampled ground-truth correspondences as an oracle upper bound. +x.x denotes improvement over DUSt3R (retrained). Best in **bold**, second best underlined.

Method	medRRE↓	RRA@5°↑	RRA@15°↑	mAA@5↑	mAA@15↑	mAA@30↑
DUSt3R [44]	24.67°	1.0%	37.0%	0.0	9.5	25.6
DUSt3R (retrained)	15.05°	10.0%	50.0%	2.4	18.8	35.7
CLICK3R w/o prompts	17.30°	13.0%	47.0%	4.6	21.5	36.2
CLICK3R (human)	8.78°	22.0%+12.0	69.0%+19.0	9.2+6.8	37.3+18.5	56.2+20.5
CLICK3R (random GT $K=4$)	**8.40°**	**27.0%**+17.0	**79.0%**+29.0	**9.4**+7.0	**39.5**+20.7	**59.2**+23.5



(a) Confidence drops as perturbation magnitude increases.

(b) Misleading hints impacting model matching performance

Figure 5: **Correspondence perturbation analysis.** (a) Model confidence at the clicked pixels decreases as perturbation strength increases (CO3D, ScanNet++, and ARKitScenes). (b) 3D distance between the predicted points at the ground-truth correspondence (ARKitScenes): the model remains robust under small perturbations, but larger perturbations degrade matching performance.

benchmarks and both Hard3DBench 2-view splits with one GT click (Appendix E.1), training from CroCo v2 performs best (mAA@30 75.1%), followed by DUSt3R finetuning (72.3%), while freezing DUSt3R gives the weakest results (71.2%). This suggests that CLICK3R requires substantial end-to-end adaptation to align backbone features with correspondence-conditioned reasoning; training the adapter alone is insufficient.

Correspondence perturbation analysis. To study how the model uses sparse correspondence hints, we perturb a single correspondence by fixing the click in view 1 and shifting its matched point in view 2 along a random direction. As perturbation increases, model confidence consistently decreases, indicating sensitivity to correspondence reliability (Fig. 5a).

We further measure whether predictions follow the true or perturbed correspondence by computing the 3D distance between predicted points at the ground-truth correspondence locations in both views. The model remains largely robust to small perturbations, but becomes more sensitive once perturbations reach half a patch (8 pixels), leading to larger degradation (Fig. 5b). For more analysis on how correspondences affects the adapters and downstream tasks, see Section F.

We also refer readers to our analyses of the effect of the number of prompts (Section E.3), robustness to image augmentation (Section E.4), and error distribution (Section E.8).

6 Conclusion

We presented CLICK3R, an interactive stereo 3D reconstruction framework that bridges fully automatic feed-forward methods, correspondence-based classical pipelines, and recent prompt-based vision systems. By injecting sparse point correspondences into a feed-forward stereo 3D reconstruction model through a lightweight correspondence adapter, CLICK3R enables users to guide the reconstruction process without test-time model fine-tuning or optimization, resolving geometric ambiguities that often cause catastrophic failures. Experiments on our curated Hard3DBench demon-

strate that even a handful of correspondences can dramatically improve reconstruction quality in challenging scenarios. While CLICK3R has limitations (Appendix J) and failure cases (Appendix H), we hope this work inspires further exploration of correspondence prompting as a promising direction for interactive 3D reconstruction, analogous to how SAM [15] advanced interactive prompting for 2D segmentation.

Acknowledgments and Disclosure of Funding

The authors acknowledge the University of Virginia Research Computing and Data Analytics Center, Advanced Micro Devices AI and HPC Cluster Program, Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, and National Artificial Intelligence Research Resource (NAIRR) Pilot for computational resources, including the Anvil supercomputer (National Science Foundation award OAC 2005632) at Purdue University and the Delta and DeltaAI advanced computing resources (National Science Foundation award OAC 2005572).

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6290–6301, 2022.

[2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

[3] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1050–1060, 2025.

[4] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1300–1309, 2022.

[5] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025.

[6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[7] Johan Edstedt, David Nördstrom, Yushan Zhang, Georg Bökman, Jonathan Astermark, Viktor Larsson, Anders Heyden, Fredrik Kahl, Mårten Wadenbäck, and Michael Felsberg. Roma v2: Harder better faster denser feature matching. In *European Conference on Computer Vision*, pages 664–682. Springer, 2026.

[8] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2003.

[9] Aaron Hertzmann. Interactive 3d scene reconstruction from images, 1999.

[10] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3r: Empowering unconstrained 3d reconstruction with camera and scene priors. pages 1071–1081, 2025.

[11] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014.

[12] Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. *arXiv preprint arXiv:2512.10935*, 2025.

[13] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.

[14] Ramil Khafizov, Artem Komarichev, Ruslan Rakhimov, Peter Wonka, and Evgeny Burnaev. G-cut3r: Guided 3d reconstruction with camera and depth prior integration. *arXiv preprint arXiv:2508.11379*, 2025.

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[16] 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.

[17] Samuel Li, Pujith Kachana, Prajwal Chidananda, Saurabh Nair, Yasutaka Furukawa, and Matthew Brown. Rig3r: Rig-aware conditioning for learned 3d reconstruction. *arXiv preprint arXiv:2506.02265*, 2025.

[18] Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Mono3r: Exploiting monocular cues for geometric 3d reconstruction. pages 11081–11090, 2025.

[19] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018.

[20] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3159–3167, 2016.

[21] Haotong Lin, Sida Peng, Jingxiao Chen, Songyou Peng, Jiaming Sun, Minghuan Liu, Hujun Bao, Jiashi Feng, Xiaowei Zhou, and Bingyi Kang. Prompting depth anything for 4k resolution accurate metric depth estimation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17070–17080, 2025.

[22] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 17627–17638, 2023.

[23] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22290–22300, 2023.

[24] Ziqi Lu, Heng Yang, Danfei Xu, Boyi Li, Boris Ivanovic, Marco Pavone, and Yue Wang. Lora3d: Low-rank self-calibration of 3d geometric foundation models. *arXiv preprint arXiv:2412.07746*, 2024.

[25] 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.

[26] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023.

[27] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[28] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021.

[29] Sara Rojas, Matthieu Armando, Bernard Ghanem, Philippe Weinzaepfel, Vincent Leroy, and Gregory Rogez. Hamst3r: Human-aware multi-view stereo 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5027–5037, 2025.

[30] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004.

[31] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[32] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.

[33] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *European Conference on Computer Vision*, pages 1–19. Springer, 2022.

[34] 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.

[35] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017.

[36] Philipp Schröppel, Jan Bechtold, Artemij Amiranashvili, and Thomas Brox. A benchmark and a baseline for robust multi-view depth estimation. In *2022 International Conference on 3D Vision (3DV)*, pages 637–645. IEEE, 2022.

[37] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8839–8849, 2024.

[38] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013.

[39] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *2022 IEEE international conference on image processing (ICIP)*, pages 3141–3145. IEEE, 2022.

[40] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5283–5293, 2025.

[41] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pages 78–89. IEEE, 2025.

[42] 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.

[43] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025.

[44] 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.

[45] 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. *arXiv preprint arXiv:2507.13347*, 2025.

[46] Ethan Weber, Riley Peterlinz, Rohan Mathur, Frederik Warburg, Alexei A Efros, and Angjoo Kanazawa. Toon3d: Seeing cartoons from new perspectives. *arXiv preprint arXiv:2405.10320*, 2024.

[47] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. CroCo v2: Improved Cross-view Completion Pre-training for Stereo Matching and Optical Flow. In *ICCV*, 2023.

[48] Weinzaepfel, Philippe and Leroy, Vincent and Lucas, Thomas and Brégier, Romain and Cabon, Yohann and Arora, Vaibhav and Antsfeld, Leonid and Chidlovskii, Boris and Csurka, Gabriela and Revaud Jérôme. CroCo: Self-Supervised Pre-training for 3D Vision Tasks by Cross-View Completion. In *NeurIPS*, 2022.

[49] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5249–5260, 2025.

[50] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22378–22389, 2024.

[51] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 373–381, 2016.

[52] Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Ekaterina Tolstaya, Sarah Tang, Brandyn White, et al. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. *arXiv preprint arXiv:2510.26125*, 2025.

[53] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025.

[54] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10371–10381, 2024.

[55] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *European conference on computer vision*, pages 785–801. Springer, 2018.

[56] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1790–1799, 2020.

[57] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023.

[58] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. 2024.

Supplementary Material

We provide implementation details and training configuration (Section A), details on Hard3DBench construction (Section C), per-category analysis on the 2-view subset (Section D), additional experiments including training strategy ablation on all evaluation sets, stricter pose thresholds, prompt count ablation, augmentation robustness, incorrect prompts, automatic prompts, MASt3R transfer, and error distribution analysis (Section E), an internal analysis of how the correspondence adapter modifies encoder features and decoder attention (Section F), additional qualitative results (Section G), failure case analysis (Section H), licenses of the existing assets we use (Section I), limitation (Section J), and broader impacts (Section K).

A Implementation Details

CLICK3R follows the DUSt3R [44] training recipe (loss function, optimizer, data augmentation, symmetric batching) and inherits its CroCo v2 [47] ViT-Large encoder and dual asymmetric decoder. Below we detail the components specific to CLICK3R.

Correspondence adapter. We insert two SAM-style [15] Two-Way Transformer adapters (depth 2, 8 heads, MLP dim 2048, embedding dim 768) between the encoder and decoder (Section 3.3). The two adapters share a single correspondence tokenizer (Section 3.2) that encodes each correspondence point as the sum of a positional encoding (random Fourier features [15]), an image identity embedding, and a pair identity embedding (vocabulary size 256). When prompts are provided, the adapted features replace the encoder output; otherwise, the adapter is bypassed and the model reduces to standard DUSt3R. The adapters and tokenizer are randomly initialized; all other weights are initialized from the CroCo pretrained checkpoint. All parameters are trained jointly end-to-end.

Training data. We train on nine datasets, each contributing 100,000 image pairs: BlendedMVS [56], MegaDepth [19], CO3D [28], StaticThings3D [36], Waymo Open [52], ARKitScenes [2], WildRGB-D [50], Habitat [32], and ScanNet++ [57]. Validation uses held-out splits from six of these datasets with 1,000 pairs each.

Ground-truth prompt generation. During training, correspondence prompts are generated by projecting view 1's 3D pointmap into view 2 and accepting correspondences where the 3D distance to the nearest view 2 surface point is below 5 cm (which implicitly rejects occluded points). Per sample, 30% receive zero prompts (adapter bypassed), 40% receive 1–4, and 30% receive 5–12 correspondences.

Training configuration. We train for 100 epochs on 4 GPUs with per-GPU batch size 16, using AdamW (β_1=0.9, β_2=0.95, weight decay 0.05), a learning rate of 10^{-4} with 10 epochs of linear warmup followed by cosine decay to 10^{-6}, and float16 mixed precision.

Baseline image preprocessing. DUSt3R-family models (DUSt3R, MASt3R, CLICK3R) use images normalized to $[-1, 1]$. The VGGT and Pi3 wrappers internally denormalize images to $[0, 1]$ before forwarding to match each model's expected input range.

Standard benchmark data. The held-out splits are the CO3D test sequences, the ScanNet++ validation scenes, and the ARKitScenes validation videos, each processed with the DUSt3R pipeline.

Evaluation metric details. The **scale-and-shift-invariant pointmap error** aligns the predicted pointmap to ground truth via an optimal global scale and shift, then averages the per-pixel L2 distance over all valid pixels (i.e., L2 within each 3D point, mean across pixels). For **mAA@θ**, relative camera pose is first estimated from each model's predicted pointmaps via PnP-RANSAC (`cv2.solvePnPRansac` with the SQPNP solver), recovering per-view camera-to-world poses. Rotation error is the geodesic distance $\arccos\big((\mathrm{tr}(R_{\mathrm{gt}}^{\top} R_{\mathrm{pred}}) - 1)/2\big)$ in degrees; translation error is the angular deviation between predicted and ground-truth translation directions, with sign ambiguity handled by $\min(\alpha, 180^\circ - \alpha)$. mAA@$\theta$ is the area under the accuracy curve of the pose error $\max(R_{\mathrm{err}}, T_{\mathrm{err}})$ up to threshold θ°. All pairs are *symmetrized* (evaluated in both directions) following the standard DUSt3R protocol.

Table 4: **Runtime and model complexity.** GPU compute time measured on a single A100 (224×224, bf16). CLICK3R without prompts has identical cost to DUSt3R; $K=8$ prompts add 12% overhead.

Configuration	Params (M)	GPU time (ms)	Overhead
DUSt3R [44] (pretrained)	532.3	30.0	—
CLICK3R (no prompts)	566.4	30.1	+0%
CLICK3R ($K=8$)	566.4	33.9	+12%
CLICK3R ($K=12$)	566.4	34.0	+13%

Runtime and Model Complexity. Table 4 reports parameter counts and GPU compute times on a single NVIDIA A100 at 224×224 resolution using bf16 mixed precision. The correspondence adapter adds 34.1M parameters (6.4%) over DUSt3R. When no prompts are provided, the adapter is bypassed entirely and CLICK3R has the same compute cost as DUSt3R (30.1 ms vs. 30.0 ms). With $K=8$ prompts, the two-way transformer adds $\sim$3.8 ms of GPU compute (12%), and this overhead is nearly constant across prompt counts since the bottleneck is the fixed-size image token processing, not the number of prompts. All times report pure GPU compute (CUDA kernel time) for the model forward pass only, excluding prompt acquisition.

B Training Details

Generating Training Correspondences. During training, we generate prompts from ground-truth 3D geometry. For each image pair with known cameras and pointmaps, we first sample the number of correspondence prompts according to a curriculum: 30% of samples use no prompts ($K=0$), 40% use a small number of prompts ($K\in[1,4]$), and 30% use more prompts ($K\in[5,12]$). Given the sampled value of K, we: (1) sample K 3D points from the first view; (2) project them into both views using known intrinsics and extrinsics; (3) retain only points that are mutually visible, verified by comparing the projected 3D position against the second view's GT pointmap within a distance threshold.

Training Protocol. We initialize from CroCo v2 [47] pretrained weights and train all parameters end-to-end, including both the base model and the adapter. Training uses 4 GPUs, AdamW ($\text{lr}=10^{-4}$) with linear warmup and cosine decay. We train on the same dataset mixture as DUSt3R: CO3D [28], ARKitScenes [2], ScanNet++ [57], WildRGB-D [50], BlendedMVS [56], MegaDepth [19], Habitat [32], Waymo [52], and StaticThings3D [36] (100K pairs each), at 224×224 resolution. With batch size 16 per GPU on A100s, a training step takes 0.60 s for CLICK3R and 0.55 s for DUSt3R (retrained), which amounts to about 234 and 215 hours for 100 epochs on 4 GPUs, with a peak memory of 29,069 and 28,731 MiB per GPU.

C Hard3DBench Details

Hard3DBench has two subsets: a *2-view stereo* subset of 404 in-distribution and 300 out-of-distribution pairs, and a *5-view multi-view* subset of 400 in-distribution and 300 out-of-distribution tuples. Both subsets use *pretrained* DUSt3R's pose error as the difficulty signal, rather than CLICK3R or our retrained baseline, so the benchmark is defined independently of the methods under evaluation. They share the same source datasets: the in-distribution split uses held-out test/val splits of four DUSt3R training datasets, while the out-of-distribution split uses five datasets entirely absent from training (Table 6 reports the 2-view per-dataset statistics). DTU uses the cameras of MVSNet [55] and the depth maps rendered by Spann3R [41], 7Scenes depth is registered to the color camera as in SimpleRecon [33], ScanNet is prepared following MonST3R [58], and ETH3D following Pi3 [45].

2-view subset. To obtain hard cases, we evaluate the pretrained DUSt3R model on up to 20,000 randomly drawn image pairs per dataset ($\sim$69K in-distribution, $\sim$68K out-of-distribution), prefiltering pairs with insufficient co-visibility via depth reprojection in DUSt3R's 224×224 crop space. We rank pairs by rotation error and select the top-N worst per dataset ($N=1{,}000$ for BlendedMVS, CO3D, WildRGB-D, DTU, and ETH3D; $N=300$ for 7Scenes, NRGBD, and ScanNet). For Habitat, only the 406 of its top 1,000 pairs from scenes not used in training are kept. This yields 3,406 in-distribution and 2,900 out-of-distribution candidates. Each candidate is then classified into one

Table 5: **Failure taxonomy for Hard3DBench.** Categories are multi-label, detected from ground-truth poses and depth.

Category	Detection criterion
Large viewpoint	Relative rotation $> 60^\circ$
Moderate viewpoint	$30^\circ <$ relative rotation $\leq 60^\circ$
Low overlap	Co-visible pixel fraction $< 20\%$
Large baseline	Baseline / median depth > 0.3
Extreme depth range	$d_{\max}/d_{\min} > 50$

Table 6: **Dataset composition of the 2-view subset of Hard3DBench.** N_{pool}: pairs after overlap filtering; N_{hard}: candidates selected by worst rotation error; N_{final}: pairs after uniform subsampling. The 5-view subset uses the same datasets; per-dataset tuple counts are uniform (top-100 in-distribution, top-60 out-of-distribution).

Dataset	Split	N_{pool}	N_{hard}	N_{final}	Scene type
In-Distribution (404 pairs, uniformly sampled from 3,406 candidates)					
BlendedMVS [56]	val	19,719	1,000	97	Synthetic multi-view
CO3D [28]	test	19,396	1,000	109	Object-centric video
Habitat [32]	val	9,871	406	100	Synthetic indoor
WildRGB-D [50]	test	19,860	1,000	98	RGB-D video
Out-of-Distribution (300 pairs, uniformly sampled from 2,900 candidates)					
7Scenes [38]	all	19,690	300	47	Indoor relocalization
DTU [11]	all	9,141	1,000	106	Multi-view objects
ETH3D [35]	all	2,162	1,000	81	Indoor/outdoor
NRGBD [1]	all	19,669	300	34	RGB-D scenes
ScanNet [6]	all	17,777	300	32	Indoor rooms

or more geometric failure categories (Table 5) from ground-truth poses and depth. We uniformly subsample 404 in-distribution and 300 out-of-distribution pairs with a fixed seed.

5-view subset. For each scene, we sample up to 30 5-frame tuples by combining stride-based windows (strides 1, 2, 5, 10, 20) with random index sampling and deduplication; tuples are kept only if all five views form a connected graph under pairwise depth-reprojection overlap $\geq 5\%$. We then run DUSt3R on all $C(5, 2)$=10 symmetrized pairs and compose them via the standard global alignment optimizer (300 iterations, msp initialization), yielding $\sim$16K in-distribution and $\sim$4K out-of-distribution candidate tuples. We rank tuples by mean pairwise rotation error and select the top-100 per in-distribution dataset (400 total) and top-60 per out-of-distribution dataset (300 total).

Difficulty for recent models. Compared with random pairs from the same datasets with the same per-dataset counts, VGGT and Pi3 at 518 resolution drop in RRA@5 from 82.6% and 85.1% to 54.7% and 57.9% on the in-distribution pairs of the 2-view subset, and from 98.2% and 97.0% to 88.7% and 81.3% on the out-of-distribution pairs, confirming that Hard3DBench remains challenging for more recent 3D reconstruction models.

D Per-Category Analysis.

Table 7 reports mAA@30° results broken down by the five failure categories. Adding sparse correspondence prompts consistently improves performance across all categories. The largest gains are observed for geometrically challenging cases such as large viewpoint changes (+12.1) and large-baseline pairs (+8.1), where automatic matching is particularly unreliable. Substantial improvements are also seen for low-overlap pairs (+6.2) and scenes with extreme depth range (+6.6), suggesting that a few well-placed correspondences help stabilize geometric estimation. Moderate viewpoint changes benefit as well (+4.3), though the baseline model already performs relatively well in this regime. Overall, prompting improves the aggregate score from 54.6 to 61.7 (+7.1), demonstrating that sparse human-provided correspondences consistently help resolve challenging geometric configurations.

Table 7: **Per-category results on the 2-view subset of Hard3DBench.** mAA@30° (↑) grouped by failure taxonomy. Categories are multi-label (pairs may belong to multiple categories). CLICK3R with $K=8$ GT correspondences. +x.x denotes improvement over DUSt3R (retrained).

Method	Lg.View.	Mod.View.	Low Ovlp.	Lg.Base.	Ext.Depth	All
DUSt3R [44]	35.9	52.0	46.3	44.6	38.9	41.8
DUSt3R (retrained)	54.3	56.3	48.7	58.0	41.3	54.6
CLICK3R w/o prompts	53.2	56.3	46.5	56.6	39.4	53.1
CLICK3R (8 prompts)	**66.4**+12.1	**60.6**+4.3	**54.9**+6.2	**66.0**+8.1	**48.0**+6.6	**61.7**+7.1

Table 8: **Ablation of training strategies on all evaluation sets** (mAA@30°, %). All variants train the correspondence adapter. Same random GT clicks for every model. H3B: Hard3DBench 2-view. Best in **bold**.

Training	CO3D	ScanNet++	ARKitScenes	H3B ID	H3B OOD	Mean
No clicks						
Adapter on frozen DUSt3R	79.2	82.6	**86.0**	39.2	47.1	66.8
Finetuned from pretrained DUSt3R	83.4	88.2	79.8	**55.5**	49.7	71.3
Trained from scratch (ours)	**84.4**	**89.9**	84.3	55.3	**50.8**	**72.9**
One click						
Adapter on frozen DUSt3R	84.0	90.1	84.8	49.7	47.2	71.2
Finetuned from pretrained DUSt3R	83.7	88.9	80.3	56.7	51.6	72.3
Trained from scratch (ours)	**85.0**	**91.1**	**85.2**	**57.5**	**56.6**	**75.1**
Eight clicks						
Adapter on frozen DUSt3R	85.3	91.0	85.2	52.4	50.4	72.9
Finetuned from pretrained DUSt3R	83.8	89.3	80.3	57.3	54.4	73.0
Trained from scratch (ours)	**85.9**	**92.2**	**87.3**	**61.8**	**62.4**	**77.9**

E Additional Experiments

E.1 Ablation of Training Strategies on All Evaluation Sets

Table 8 extends the training strategy ablation of Section 5 to all five evaluation sets, with the same random GT clicks for every model.

E.2 Stricter Pose Thresholds

Tables 9 and 10 extend Tables 1 and 2 with mAA@5° and mAA@15°. Against DUSt3R (retrained), the gain grows at stricter thresholds on the standard benchmark (grid oracle, $+15.7$ at 5° vs. $+5.7$ at 30°), where prompts sharpen reasonable poses, and at looser thresholds on Hard3DBench ($K=8$, $+3.6$ vs. $+7.7$), where prompts mainly rescue large failures.

E.3 Effect of the Number of Prompts

Figure 6 shows pose accuracy as a function of the number of correspondence prompts K on Hard3DBench. We vary $K \in \{0, 1, 2, 4, 8, 12\}$, where $K = 0$ denotes the unprompted mode in which the adapter is bypassed entirely. All methods use 224×224 input for a fair comparison.

Accuracy generally improves with K on both splits. The effect is more pronounced on the out-of-distribution split, where even a single click ($K=1$) yields a substantial gain. The improvement is larger on the out-of-distribution split ($+11.7$ pp RRA@15 from $K=0$ to $K=8$) than in-distribution ($+8.8$ pp), consistent with the base model having more room for improvement on unseen data. With $K \geq 2$, CLICK3R outperforms both DUSt3R variants and VGGT on both RRA@15 and mAA@30. On the OOD split, Pi3 [45] remains stronger at every K, likely due to its larger training data and model capacity.

Table 9: **Pose accuracy at stricter thresholds on the standard benchmark.** mAA at 5°, 15° and 30° for every row of Table 1, on the same pairs; the mAA@30° columns equal Table 1. Gray rows are reference only and not ranked. ±x.x denotes the difference from DUSt3R (retrained). Best in **bold**, second best underlined.

	CO3D			ScanNet++			ARKitScenes		
Method	@5°	@15°	@30°	@5°	@15°	@30°	@5°	@15°	@30°
VGGT [42] (518)	55.67	79.22	87.76	21.80	44.03	59.47	34.60	63.85	77.91
Pi3X [45] (518)	54.17	79.26	87.95	47.80	69.68	80.77	40.64	69.55	82.16
VGGT [42] (224)	4.37	36.78	62.53	0.20	7.14	24.47	1.26	17.37	40.89
Pi3X [45] (224)	44.27	73.85	85.02	31.80	57.70	72.53	31.68	61.41	76.39
DUSt3R [44]	33.25	65.17	79.20	50.24	73.83	82.57	48.46	75.15	86.02
MASt3R [16]	31.38	62.86	78.45	7.12	29.93	51.79	17.70	47.11	65.40
DUSt3R (retrained)	45.63	73.53	84.33	60.02	83.11	90.81	42.48	70.23	82.71
CLICK3R w/o prompts	46.12+0.5	73.43−0.1	84.41+0.1	59.64−0.4	82.05−1.1	89.87−0.9	46.18+3.7	73.03+2.8	84.27+1.6
CLICK3R (1p_best5)	47.20+1.6	74.92+1.4	85.71+1.4	62.88+2.9	84.43+1.3	91.75+0.9	48.24+5.8	75.01+4.8	85.90+3.2
CLICK3R (grid oracle)	**53.62**+8.0	**78.79**+5.3	**88.21**+3.9	**72.10**+12.1	**88.57**+5.5	**93.94**+3.1	**69.66**+27.2	**86.49**+16.3	**92.75**+10.0

Table 10: **Pose accuracy at stricter thresholds on Hard3DBench.** mAA at 5°, 15° and 30° for every row of Table 2, over finite poses; 5-view values pool the ten pairwise poses of each tuple. The mAA@30° columns equal Table 2. Gray rows are reference only and not ranked. ±x.x denotes the difference from DUSt3R (retrained). Best in **bold**, second best underlined.

	2-view (pair)			*5-view (tuple)*		
Method	@5°	@15°	@30°	@5°	@15°	@30°
In-Distribution (404 pairs / 400 tuples)						
VGGT [42] (518)	27.4	50.1	63.7	32.5	48.5	56.5
Pi3 [45] (518)	34.3	53.1	64.5	41.0	54.4	60.3
VGGT [42] (224)	3.5	19.1	40.5	2.0	16.9	33.2
Pi3 [45] (224)	22.9	46.7	60.7	30.1	48.7	56.9
DUSt3R [44]	2.3	18.3	39.2	5.7	14.5	21.1
MASt3R [16] (512)	**21.8**	**46.1**	**62.9**	19.4	38.4	48.4
DUSt3R (retrained)	15.3	38.4	56.1	15.6	30.6	39.4
CLICK3R w/o prompts	15.7	37.9	55.3	15.8	30.6	38.8
CLICK3R (1 prompt)	16.5+1.2	39.6+1.1	57.5+1.3	17.6+2.0	35.4+4.8	45.0+5.6
CLICK3R (4 prompts)	19.0+3.7	43.9+5.5	61.3+5.2	**19.9**+4.3	38.6+8.0	48.6+9.2
CLICK3R (8 prompts)	19.0+3.6	44.1+5.7	61.8+5.7	**19.9**+4.3	**39.1**+8.5	**49.3**+9.9
Out-of-Distribution (300 pairs / 300 tuples)						
VGGT [42] (518)	52.1	72.0	81.3	49.5	70.5	81.1
Pi3 [45] (518)	41.2	68.9	80.6	44.6	69.9	81.4
VGGT [42] (224)	4.9	26.0	47.4	4.4	25.4	47.1
Pi3 [45] (224)	24.1	55.6	71.1	33.3	60.2	73.5
DUSt3R [44]	9.1	27.3	45.3	15.9	40.2	54.7
MASt3R [16] (512)	14.4	33.2	51.6	**19.6**	42.4	58.0
DUSt3R (retrained)	10.3	36.2	52.6	14.4	39.0	54.8
CLICK3R w/o prompts	11.1	36.0	50.8	13.9	38.9	54.7
CLICK3R (1 prompt)	12.7+2.3	40.3+4.0	56.6+4.0	14.2−0.2	40.0+1.0	56.7+1.9
CLICK3R (4 prompts)	14.2+3.9	44.1+7.8	60.4+7.8	15.2+0.8	41.7+2.8	58.6+3.8
CLICK3R (8 prompts)	**15.3**+5.0	**45.2**+8.9	**62.4**+9.8	16.0+1.6	**43.1**+4.1	**60.0**+5.2

E.4 Robustness to Image Augmentations

We test how well CLICK3R tolerates photometric perturbations by augmenting only view 2 while keeping view 1 unchanged, and measuring RRA@15° on 100 hard OOD pairs. We consider two augmentation types at two severity levels each: brightness scaling ($\times 0.5$, $\times 1.5$) and additive Gaussian noise ($\sigma \in \{0.1, 0.3\}$).

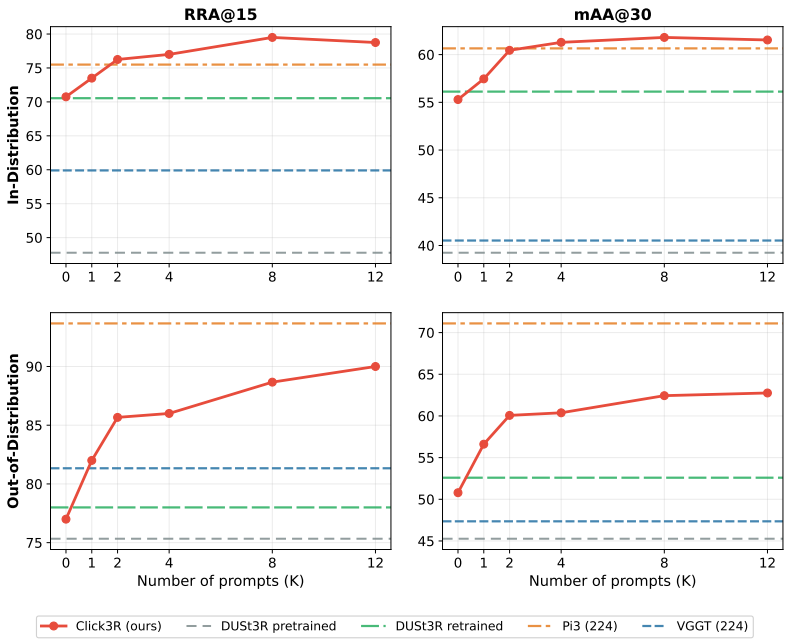


Figure 6: **Prompt count ablation on Hard3DBench.** Pose accuracy of CLICK3R as a function of K (red), with baselines shown as horizontal lines. Top: in-distribution (404 pairs); bottom: out-of-distribution (300 pairs). All methods use 224×224 resolution.

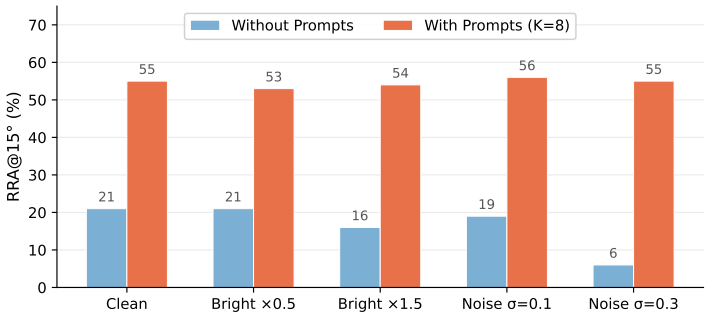


Figure 7: **Robustness to photometric augmentations.** We perturb view 2 with brightness scaling or Gaussian noise and report RRA@15° on 100 hard OOD pairs. Prompts consistently improve accuracy, with the largest margin under heavy noise (σ=0.3): 55% vs. 6%.

As shown in Figure 7, brightness changes have negligible impact on both the unprompted and prompted models. Gaussian noise, however, exposes a clear robustness gap. At σ=0.3 the unprompted model drops from 21% to 6% RRA@15°, whereas the prompted model stays at 55%, suggesting that correspondence prompts provide geometric constraints that compensate for corrupted visual features.

E.5 Robustness to Incorrect Prompts

Table 11 complements the perturbation analysis of Section 5.5 with completely incorrect clicks: pose accuracy degrades monotonically as incorrect clicks replace correct ones, and all-incorrect clicks fall below using no prompts.

E.6 Automatic Prompts

We also consider scenarios without ground-truth correspondences, where we use automatic prompting from RoMa v2 [7], a state-of-the-art dense feature matcher. We run it on each image pair, rank its

Table 11: **Robustness to incorrect prompts.** On the 100 pairs of Table 3, we use 8 random ground-truth clicks and replace a fraction of the view-2 clicks by locations sampled uniformly over the image, at least 16 pixels from the true correspondence. Results are pooled over 5 random draws per pair.

Correct clicks (of 8)	medRRE↓	mAA@30↑
100%	7.70°	62.8
75%	8.81°	55.6
50%	12.08°	44.9
25%	20.63°	30.0
0%	34.76°	17.7
No prompts	18.64°	35.6

Table 12: **Automatic prompts from RoMa v2 [7].** mAA@30° of CLICK3R with clicks from RoMa v2 matches or from random ground-truth correspondences (GT). H3B: Hard3DBench 2-view (404 ID, 300 OOD pairs); the subset is the 100 OOD pairs with human clicks.

	H3B ID		H3B OOD		H3B OOD subset	
Clicks	RoMa v2	GT	RoMa v2	GT	RoMa v2	GT
$K=0$	55.9		49.4		35.6	
$K=1$	58.4	57.8	57.0	55.8	48.7	48.1
$K=4$	58.0	60.5	59.6	60.8	55.0	58.7
$K=8$	58.4	62.4	59.4	61.8	58.5	61.1
Human	–		–		56.8	

matches by confidence, and greedily select K of them with a spacing of 10% of the image width in both views, so that the clicks spread over the scene. These clicks improve mAA@30 on all three sets of Table 12: with $K=4$, from 55.9 to 58.0 on the in-distribution pairs, from 49.4 to 59.6 on the out-of-distribution pairs, and from 35.6 to 55.0 on the 100 out-of-distribution pairs with human clicks, close to the human clicks themselves (56.8).

E.7 Transfer to MASt3R

As a controlled study of whether correspondence prompting transfers to another backbone, we add the adapter to MASt3R [16] and train it end-to-end, along with a control without the adapter on the same data. Both are trained on the same nine datasets as CLICK3R (CO3D, ARKitScenes, ScanNet++, WildRGB-D, BlendedMVS, MegaDepth, Habitat, Waymo, and StaticThings3D), a smaller mixture than the official MASt3R recipe, so their absolute numbers are not comparable with the released model. Test-time clicks raise mAA@30 from 61.1 to 62.4 on the in-distribution pairs and from 52.1 to 54.9 on the out-of-distribution pairs (Table 13), confirming that correspondence clicks also improve performance on a different backbone.

E.8 Error Distribution Analysis

We plot per-sample distributions of reconstruction loss and pose error across the three evaluation datasets of Table 1, comparing CLICK3R against DUSt3R [44], MASt3R [16], VGGT [42], and Pi3X [45] at 224×224 resolution.

Reconstruction loss. Figure 8 shows the distribution of the scale-and-shift-invariant pointmap loss. Both CLICK3R variants (with and without prompts) have more mass near zero than DUSt3R: the fraction of samples with loss >1.0 drops from 14.9% (DUSt3R) to 11.4% (CLICK3R), and with loss >2.0 from 6.1% to 3.8%. CLICK3R without prompts already accounts for most of this reduction. VGGT [42] at 224×224 shows a broader distribution (63.0% of samples exceeding loss 1.0), reflecting its sensitivity to input resolution.

Pose error. Figure 9 shows the distribution of the combined pose error $\max(R_{\text{err}}, T_{\text{err}})$. CLICK3R with prompts produces a sharper peak near 0° and a shorter tail than all baselines. The fraction of

Table 13: **Correspondence prompting on MASt3R.** mAA@30° on Hard3DBench 2-view at 512 resolution (404 ID, 300 OOD pairs). $K=8$ uses random ground-truth clicks.

	H3B ID	H3B OOD
MASt3R (retrained)	61.3	50.6
MASt3R + adapter, $K=0$	61.1	52.1
MASt3R + adapter, $K=8$	62.4	54.9

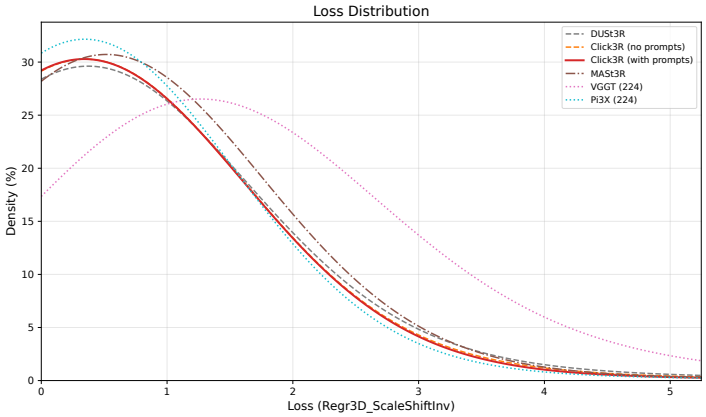


Figure 8: **Reconstruction loss distribution** (aggregated over the three datasets of Table 1). Density estimated via KDE with Scott's bandwidth. CLICK3R concentrates more mass near zero and exhibits a shorter tail than DUSt3R.

pairs with pose error exceeding 15° drops from 9.4% (DUSt3R) to 6.1% (CLICK3R without prompts) and further to 2.8% (CLICK3R with prompts)—a 3.4× reduction. At the 30° threshold, the failure rate decreases from 4.7% to 0.8%. The reduction is concentrated in the tail rather than a uniform shift of the distribution, indicating that prompts primarily resolve the hardest cases.

F What Does the Adapter Do?

We examine how the correspondence adapter modifies encoder features and how this affects downstream decoder attention.

Feature modification. Figure 10 visualizes the relative change in each token's feature vector before and after the adapter ($\|\Delta f\|/\|f\|$) for a representative pair with $K=8$ prompts. The adapter does not apply a uniform transformation: feature changes are largest near the prompt locations (cyan dots), although every token is modified.

Effect on decoder cross-attention. Figure 11 shows how prompts reshape the decoder's cross-view attention. For three query patches in view 1 (green crosses), we visualize where each query attends in view 2 with and without prompts. Without prompts, cross-attention is diffuse and sometimes focuses on incorrect regions. With prompts, attention shifts toward the geometrically correct correspondences.

G More Qualitative Visualizations

Figure 13 presents additional qualitative comparisons on challenging pairs. For each example, we show the input pair with correspondence clicks, the ground-truth reconstruction, and the predictions of DUSt3R, Pi3, and CLICK3R. These cases illustrate scenarios where at least one baseline produces a noticeably incorrect reconstruction. For instance, DUSt3R may fail on a pair where Pi3 succeeds, or vice versa. CLICK3R with sparse correspondence clicks consistently recovers geometry close to the ground truth across all shown examples. The clicked correspondences converge to consistent 3D

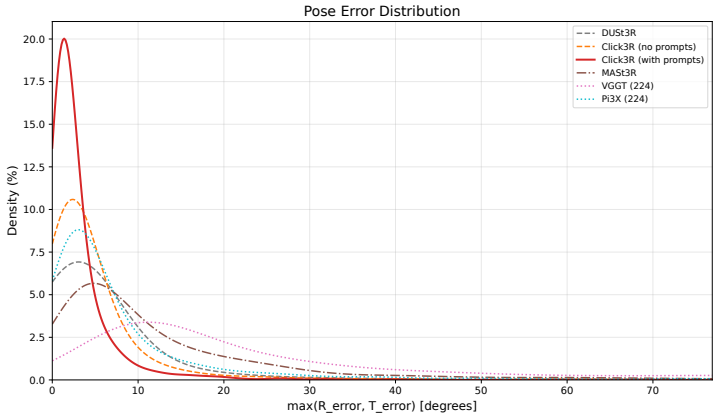


Figure 9: **Pose error distribution** (aggregated over the three datasets of Table 1 at 224×224, $\max(R_{\text{err}}, T_{\text{err}})$ in degrees). CLICK3R with prompts produces the sharpest peak near 0° and the shortest tail. The fraction of pairs exceeding 15° error drops from 9.4% (DUSt3R) to 2.8% (CLICK3R with prompts), a $3.4\times$ reduction in pose failures.

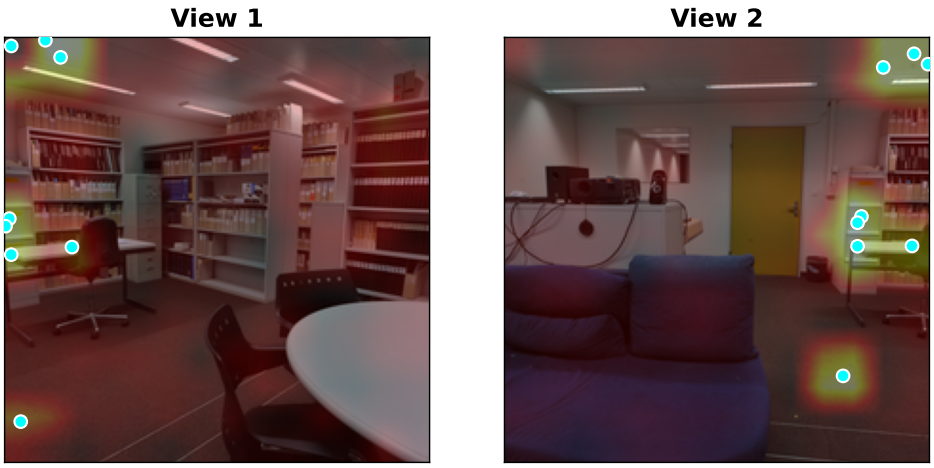


Figure 10: **What the adapter changes.** Per-token relative feature change induced by the adapter, with $K = 8$ prompt locations shown as cyan dots. Bright regions indicate tokens whose features were modified most (each view uses its own color scale). The adapter updates every token, most strongly near the prompt correspondences.

locations in the CLICK3R reconstruction, confirming that the adapter successfully leverages sparse correspondences to resolve geometric ambiguities that affect different baselines in different ways.

H Failure Cases

Figure 14 shows four representative failure modes of CLICK3R. Each panel shows the input image pair (left), the ground-truth reconstruction (middle), and the CLICK3R reconstruction with clicks (right).

(a) Few overlap. When the two views share little or no visual overlap, the model must hallucinate the relative geometry from appearance alone. In this ScanNet++ example the views observe disjoint parts of a room; comparing the ground truth with the CLICK3R output reveals a severely misaligned reconstruction. Without co-visible content, correspondence clicks cannot anchor the two views—this is a fundamental limitation of pairwise reconstruction.

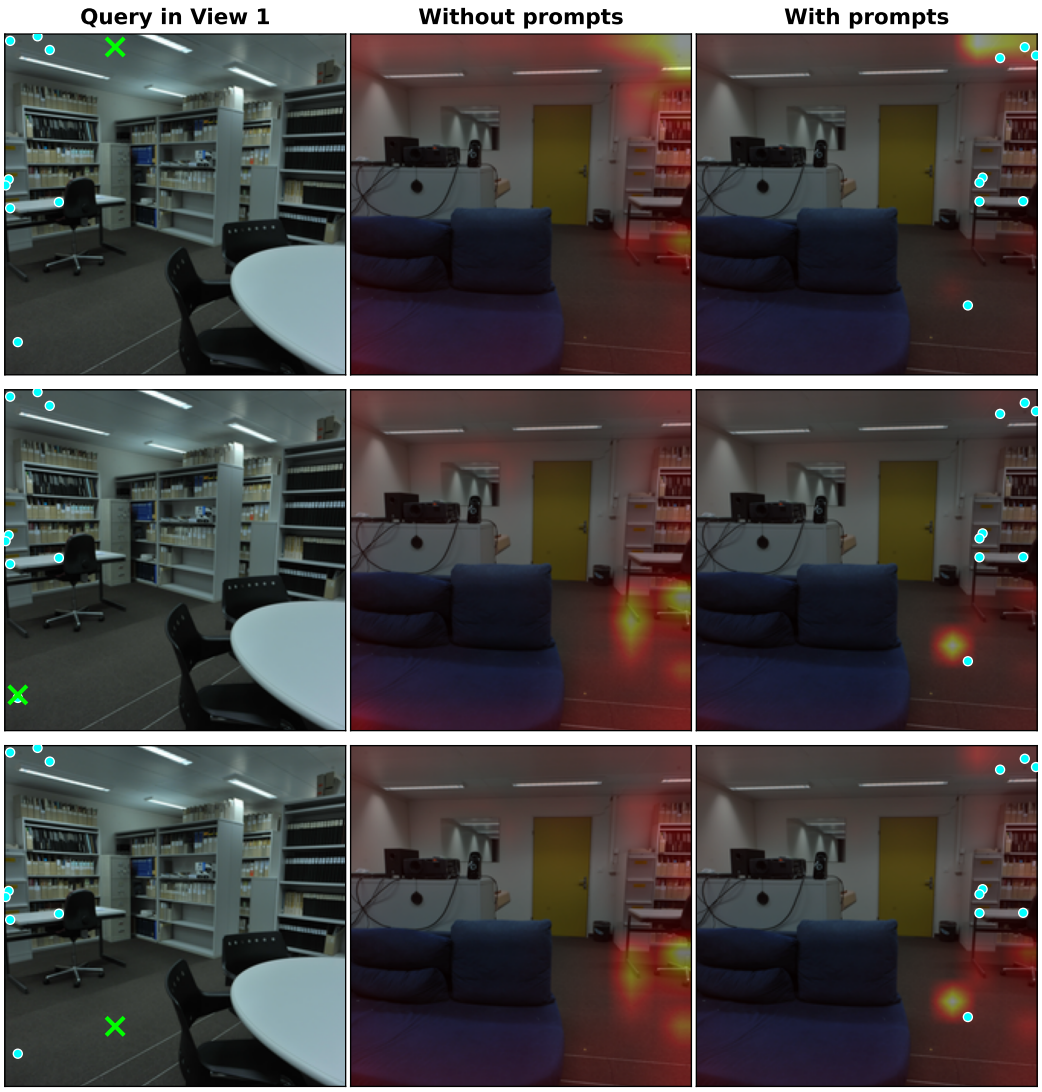


Figure 11: **How prompts reshape decoder cross-attention.** Each row shows a query patch in view 1 (green cross) and its decoder cross-attention map over view 2. Without prompts (middle), attention is diffuse. With $K = 8$ prompts (right, cyan dots), attention concentrates on the correct corresponding region.

(b) Heavy rotation. Extreme viewpoint changes ($> 90°$) remain an unsolved challenge. In this ARKitScenes pair, the CLICK3R reconstruction remains geometrically distorted compared to the ground truth. While clicks can reduce error in moderate-rotation cases (Table 7), very heavy rotations overwhelm the model's capacity to reconcile drastically different visual appearances.

(c) Large-scale scenes. For outdoor scenes with large spatial extent (MegaDepth), correspondence clicks correctly constrain geometry near the clicked regions, but distant structures still exhibit bending and deformation—visible here as warped sky and distorted building facades far from the click locations. A small number of clicks cannot regularize the entire scene when the depth range spans tens of meters.

(d) Ambiguous correspondences. On textureless or repetitive surfaces, correspondences themselves become ambiguous. In this ARKitScenes bathroom scene with featureless marble walls and white surfaces, a click on one location is visually indistinguishable from many others, providing little geometric information. The CLICK3R reconstruction remains distorted compared to the ground truth, and clicks offer negligible improvement.

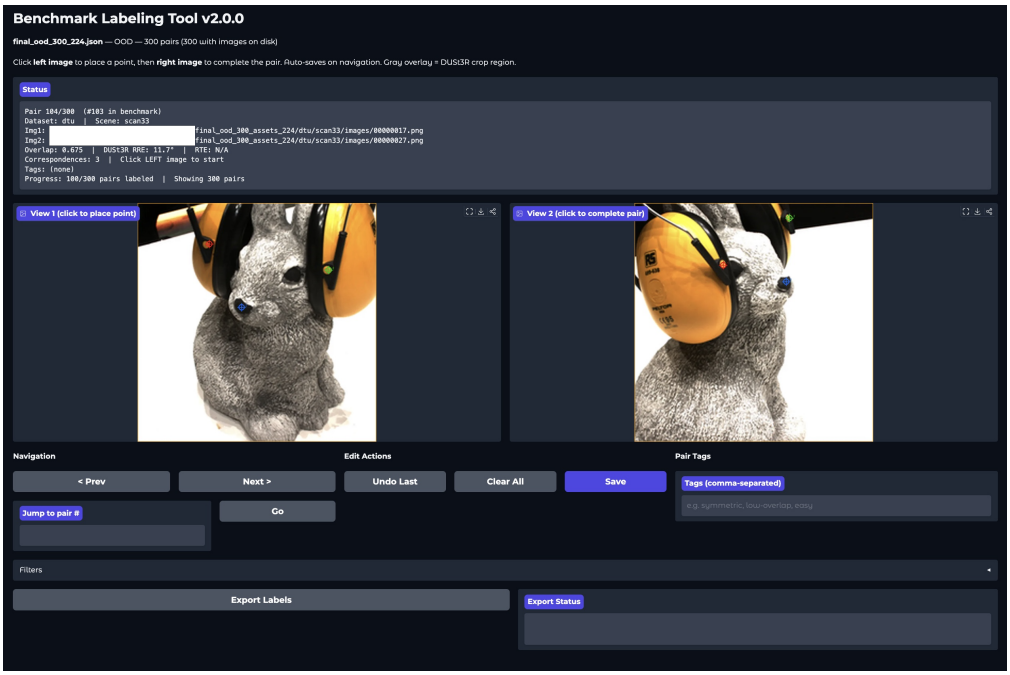


Figure 12: A screenshot of the **labeling tool** we build to harness human labeled correspondences.

Table 14: **Licenses of assets used.**

Asset	License
Datasets (training)	
CO3D [28]	CC-BY-NC 4.0
ARKitScenes [2]	Apple License (non-commercial primary)
ScanNet++ [57]	ScanNet++ Terms of Use
WildRGB-D [50]	MIT License
BlendedMVS [56]	CC-BY 4.0
MegaDepth [19]	CC-BY 4.0 (depth, SfM); images under their Flickr licenses
Habitat-Sim [32]	MIT License
Waymo Open Dataset [52]	Waymo Dataset License (non-commercial)
Datasets (Hard3DBench evaluation)	
7Scenes [38]	Microsoft Research License (non-commercial)
DTU [11]	No formal license; cite the original paper
ETH3D [35]	CC-BY-NC-SA 4.0
NRGBD [1]	CC-BY / CC-0 (per synthetic scene)
ScanNet [6]	ScanNet Terms of Use (data); MIT License (code)
Code / Models	
DUSt3R [44]	CC-BY-NC-SA 4.0
CroCo v2 [47]	CC-BY-NC-SA 4.0
Segment Anything (SAM) [15]	Apache License 2.0
MASt3R [16]	CC-BY-NC-SA 4.0
VGGT [42]	VGGT License (custom; weights non-commercial)
Pi3 [45]	BSD 3-Clause (code); CC-BY-NC 4.0 (weights)

I Licenses of Existing Assets

We acknowledge and respect the licenses of all assets used in this work. Table 14 lists the license of each dataset, code base, and pretrained checkpoint we rely on. All assets are used in compliance with their license terms, and our work is intended for non-commercial research purposes consistent with these licenses.

J Limitations

Although correspondence prompts are ideally injected into a frozen pretrained model, we find that training the base model [44] end-to-end with the adapter is necessary in practice, incurring nontrivial training cost. Partly due to this cost, our experiments focus on two-view reconstruction; extending to multi-view models [45, 42, 53, 40] with trajectory prompts across N views is an important direction.

Architecturally, the adapter treats correspondences as soft feature conditioning via cross-attention rather than enforcing hard geometric constraints. Although eight correspondences suffice to determine relative pose algebraically [8], the adapter cannot replicate such reasoning: it modulates all image tokens through learned attention at a single injection point, and this signal is progressively diluted across twelve subsequent decoder cross-attention blocks. Deeper injection at multiple decoder layers and explicit geometric reasoning modules may be worth future inquiry to help close this gap.

Finally, due to computational constraints, the model is trained only at a resolution of 224px in a stereo setting, although it has demonstrated multi-view capability through global alignment. In addition, prompts must be co-visible in both views, so the two inputs need a shared visible region; non-overlapping inputs could be connected through intermediate views by chaining correspondences. Severely incorrect correspondences can degrade reconstruction quality (Table 11).

K Broader Impacts

Our work enables more efficient 3D reconstruction from sparse user-provided correspondences. By allowing a small number of clicks to anchor pairwise reconstruction, CLICK3R can lower the barrier to 3D capture in diverse, in-the-wild scenes, which may benefit downstream applications such as robotics, AR/VR content creation, and scientific imaging where dense ground-truth geometry is unavailable. Our model and benchmark are built from publicly available datasets (CO3D, ScanNet++, ARKitScenes, BlendedMVS, MegaDepth, Habitat, WildRGB-D, Waymo, and standard evaluation benchmarks), and therefore do not raise privacy concerns. While our algorithm itself is scene-agnostic and does not introduce explicit bias, the underlying training datasets may reflect societal or geographic biases present in the source data (e.g., predominantly indoor scenes from North American buildings, or object categories skewed toward consumer products). We encourage future work to investigate and mitigate such biases when deploying systems trained on these data.

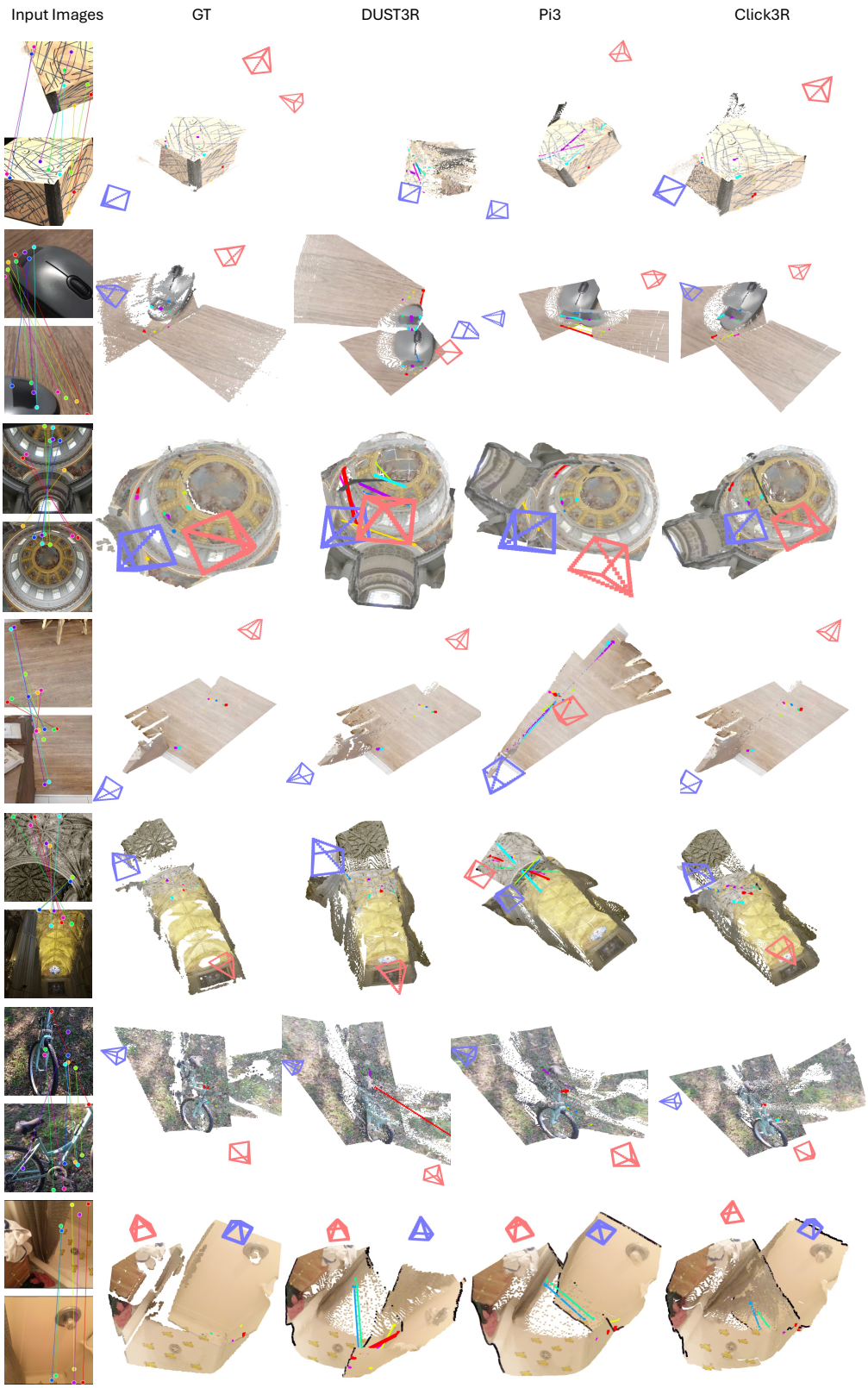
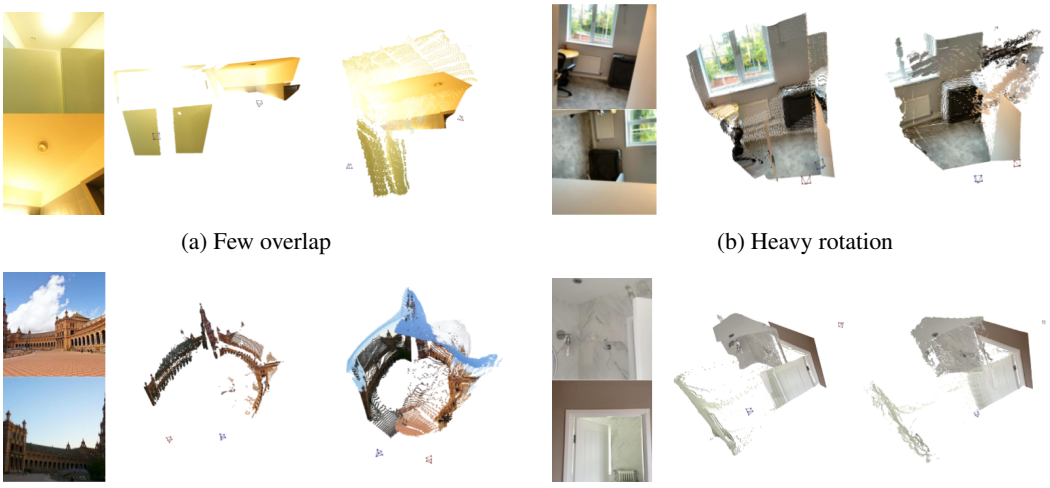


Figure 13: **Additional qualitative comparisons.** On challenging pairs, different baselines fail in different ways: DUSt3R may produce a distorted reconstruction on a pair where Pi3 succeeds, or vice versa. CLICK3R with sparse correspondence clicks consistently recovers accurate geometry.



(a) Few overlap

(b) Heavy rotation

(c) Large-scale scene

(d) Ambiguous correspondences

Figure 14: **Failure cases of CLICK3R.** Each panel: input pair (left), ground truth (middle), CLICK3R with clicks (right). (a) Near-zero overlap showcasing fundamental limitation of click-based supervision. (b) Extreme rotations cause geometric distortion that clicks cannot resolve. (c) In large-scale scenes, regions far from clicked correspondences still deform. (d) Clicks on featureless surfaces are inherently ambiguous and provide insufficient geometric signal.

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: We clearly state the scope of our work in both the abstract and the introduction. Additionally, we highlight the main contributions of our paper at the end of the introduction.

   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: We discuss the limitations of our work in the conclusion section (Future Work and Limitations) and provide an extended analysis of failure cases in the Appendix.

   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: The paper presents an empirical method for correspondence-prompted 3D reconstruction and does not include theoretical results.

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: In the experiments section, we describe the datasets, training mixture, evaluation protocols, and metrics used. The supplementary material further details the architecture, ground-truth prompt generation, training configuration, and benchmark construction. Following these descriptions, the main results and ablation studies can be reproduced.

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: We will publicly release our code, model checkpoints, and the constructed hard benchmark on our project page upon acceptance.

   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: We describe the experimental setup, datasets, evaluation protocols, and main hyperparameters in the experiments section, and provide full implementation details (architecture, optimizer, learning rate schedule, training data mixture, ground-truth prompt generation, and benchmark construction) in the supplementary material.

   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: Due to limited computing resources, we are unable to repeat the full training and evaluation pipeline multiple times to report error bars or statistical significance tests.

   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: We include the details about the computing resources used for our experiments (4 GPUs, AdamW optimizer with mixed-precision training) in the Appendix.

   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 conform with the NeurIPS Code of Ethics in every aspect.

   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: [Yes]

    Justification: We include a discussion of potential broader societal impact in the Appendix.

    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: To the best of our knowledge, no such risk is associated with the work.

    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: The papers of existing code, models, and datasets are adequately credited and cited throughout the paper. We have reviewed and complied with the licenses of all referenced works. We include a section listing the licenses of the assets used in the supplementary material.

    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: We will release all new assets (code, model checkpoints, and Hard3DBench) along with proper documentation upon acceptance.

    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: The paper does not involve crowdsourcing platforms or research with human subjects. All evaluations and ablations are conducted on existing public datasets.

    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: The paper does not involve crowdsourcing or research with human subjects, and therefore does not require IRB or equivalent ethical approval.

    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: The core methodology of this work, correspondence-prompted 3D reconstruction, does not involve LLMs as any important, original, or non-standard component. LLMs were used only for incidental writing assistance, which falls outside the scope of the declaration requirement per the NeurIPS LLM policy.

    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.