FLOWMATCHER: HETEROSCEDASTIC CONDITIONAL TRANSPORT FOR DENSE CORRESPONDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

A dense matcher should report not only where each pixel corresponds, but how reliable that correspondence is. Coarse-to-fine matchers expose this reliability at the coarse stage, where matching scores spread across candidate regions, yet decode a single warp and refine it deterministically. We introduce FlowMatcher, a stochastic refiner that makes this coarse ambiguity the source of a transport process: candidate target coordinates are drawn from a Gaussian centered on the coarse warp, with a per-pixel scale predicted from the coarse matching scores, and a time-conditioned multiscale refiner moves each candidate toward the correct correspondence. Samples converge where the evidence is decisive and stay apart where it is not, so their disagreement serves as an uncertainty signal. Built on the frozen RoMaV2 coarse matcher, FlowMatcher remains competitive with RoMaV2 in accuracy, while its sample spread ranks errors better than a RoMaV2-style precision head trained on our architecture, improves relative pose as residual weights, and yields held-out 90% regions after split-conformal calibration. The samples also contain correct answers that a single prediction misses: an oracle choosing among them raises one-pixel accuracy by 5–10 points over their mean, and a ground-truth-free cycle-consistency criterion recovers part of this gain in dense accuracy and pose. Although both refiners are trained at 640×640, when both methods are run directly at 1024×1024, FlowMatcher exceeds RoMaV2 in one-pixel accuracy on all dense benchmarks tested.

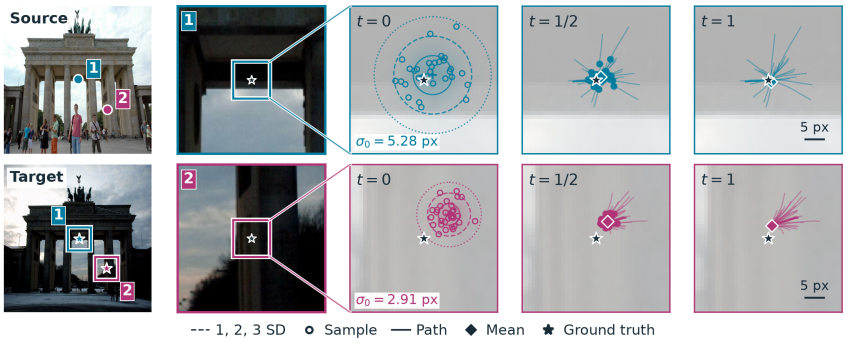


Figure 1: **Heteroscedastic transport for dense correspondence.** For two source pixels, FlowMatcher initializes candidate target coordinates around the coarse prediction, with spatially varying Gaussian scales derived from coarse matching evidence. Enlarged target neighborhoods show the samples evolving from $t = 0$ to $t = 1$.

1 INTRODUCTION

Dense feature matching establishes pixel-wise correspondences between two views of the same scene. It is a core primitive in Structure-from-Motion (Snavely et al., 2008; Schönberger & Frahm, 2016), SLAM (Mur-Artal et al., 2015), and visual localization (Sarlin et al., 2019; Lynen et al., 2020). Dense matchers must be robust to large appearance changes and localize precisely. Coarse-to-fine methods address these requirements separately: a global matcher identifies candidate regions

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and local refinement recovers spatial detail (Edstedt et al., 2023; 2024; 2026). This decomposition raises two questions. Once global matching has narrowed the search, how should the information it leaves unresolved enter refinement? And can that information tell us how reliable the final correspondence is?

Ideally, a matcher would represent the conditional distribution of correspondences given the image pair. For a well-determined pixel this distribution is sharply peaked; under repetitive patterns, weak texture, occlusion, or limited resolution, several correspondences remain compatible with the evidence (Truong et al., 2023; Edstedt et al., 2024). Coarse matching exposes part of this ambiguity: before a correspondence is decoded, matching scores spread support across candidate regions, and RoMa (Edstedt et al., 2024) argues that coarse conditionals can be multimodal near discontinuities. State-of-the-art matchers nevertheless decode this evidence into a single warp and refine it deterministically (Edstedt et al., 2023; 2024; 2026). Where correspondence is hardest, near edges, under strong viewpoint change, or in weakly constrained regions, the coarse estimate may admit several corrections, yet a deterministic refiner returns one and gives no direct indication of how strongly the image evidence determines it.

We introduce FlowMatcher, which turns the coarse evidence into the source of a transport process (Fig. 1). Candidate correspondences are drawn from a Gaussian centered on the coarse warp, whose per-pixel scale is predicted from the coarse matching scores, so refinement stays local where the coarse evidence is sharp and explores where it is diffuse. A time-conditioned refiner, trained with a flow-matching-style endpoint objective (Lipman et al., 2023; Liu et al., 2023), transports each candidate toward the fine-scale evidence it queries at its current location, following the correlation lookup of PWC-Net and RAFT (Sun et al., 2018; Teed & Deng, 2020). Because this lookup depends on the hypothesis, candidates that start in different places can see different evidence and settle on different explanations of an ambiguous region. Since the refiners are supervised against a single ground-truth warp (Sec. 3.2), the resulting endpoints measure the sensitivity of refinement to the starting hypothesis under fixed image evidence. We treat their disagreement as an empirical measure of how strongly the fine-scale evidence pins down each correspondence, and test this measure directly. Our contributions are:

- **Heteroscedastic conditional transport.** We formulate dense refinement as transport from a coarse-informed source whose mean is the coarse warp and whose per-pixel scale is predicted from coarse matching statistics, refined by a time-conditioned endpoint predictor that re-queries local evidence for every hypothesis (Sec. 3). Our model provides deterministic, averaged, and sampled predictions, and the number of integration steps N and draws K trade compute for accuracy and improved uncertainty estimation.
- **Uncertainty from sampling.** Sample spread ranks correspondence errors better than the RoMaV2-style trained precision head on ScanNet++ v2 and Hypersim (23% and 27% lower AUSE) and matches it on MegaDepth. Used as residual weights, it improves relative pose on all three pose benchmarks, and a split-conformal scale yields held-out 90% coverage (Secs. 4.2–4.3).
- **Hypotheses beyond the mean.** The samples contain correspondences that the mean misses: a ground-truth oracle over them raises PCK@1 by 5–10 points. A ground-truth-free inverse-consistency criterion recovers part of this headroom, improving PCK@1 over the mean on dense benchmarks tested. (Secs. 4.1–4.2).
- **Robustness to input resolution.** Both refiners are trained at 640×640, but FlowMatcher degrades less at higher input resolution. At native 1024×1024 it exceeds RoMaV2 in PCK@1 on all dense benchmarks tested. (Secs. 4.1–4.2).

2 Related Work

Dense correspondence and iterative refinement. Detector-free matchers such as LoFTR (Sun et al., 2021) replaced keypoint pipelines (DeTone et al., 2018; Lindenberger et al., 2023) with transformers. DKM (Edstedt et al., 2023) combines a global kernel-regression matcher with local refinement; RoMa (Edstedt et al., 2024) adds robust foundation-model features and coarse anchor probabilities that can represent competing hypotheses; RoMaV2 (Edstedt et al., 2026) decouples matching from refinement and predicts pixel-wise residual covariance. BEAMER (Vilain et al.,

2025) addresses ambiguity by preserving multiple discrete correspondence hypotheses across scales with beam search and hypothesis-aware cross-attention. In contrast, FlowMatcher obtains hypotheses as stochastic endpoints of a continuous coarse-conditioned transport process and uses their disagreement as an uncertainty signal. UFM (Zhang et al., 2025) unifies optical flow and wide-baseline matching through direct high-resolution regression, and DUSt3R/MASt3R (Wang et al., 2024; Leroy et al., 2024) obtain matches from 3D pointmap regression. Iterative optical-flow methods refine around evolving estimates: PWC-Net (Sun et al., 2018) warps target features by the current flow, and RAFT (Teed & Deng, 2020) repeatedly queries correlation volumes at updated coordinates. FlowMatcher uses an equivalent estimate-dependent lookup, but applies it to many hypotheses drawn around the coarse warp instead of to a single deterministic estimate.

Generative models for dense geometric prediction. Diffusion and flow models have been applied to monocular depth (Saxena et al., 2023; Ke et al., 2024; Gui et al., 2025; Xu et al., 2026) and optical flow (Saxena et al., 2023; Luo et al., 2024; 2026; Zuo et al., 2026). DDVM (Saxena et al., 2023) also introduced step-unrolled denoising and Monte Carlo uncertainty for these tasks. For matching, DiffMatch (Nam et al., 2024) applies conditional diffusion to dense correspondence, conditioning on an initial field and matching costs; DiffGlue (Zhang & Ma, 2024) and Diff-Reg (Wu et al., 2024) diffuse over sparse assignment and doubly stochastic matching matrices. These methods start from generic noise and supply the coarse estimate as conditioning. FlowMatcher instead places the coarse matcher at the source of the process: its warp sets the mean of the initialization and its matching evidence sets a spatially varying scale.

Correspondence uncertainty. Ilg et al. (2018) predict multiple optical-flow hypotheses and evaluate uncertainty by sparsification, which we adopt. PDC-Net and PDC-Net+ (Truong et al., 2021; 2023) predict constrained mixture densities over dense correspondences, and RoMaV2 (Edstedt et al., 2026) regresses a residual precision matrix per pixel. Sampling-based alternatives use disagreement between ensemble members (Lakshminarayanan et al., 2017) or repeated generative draws (Saxena et al., 2023; Nam et al., 2024); recent work links flow-matching dynamics to predictive variance along the trajectory (de Kruiff et al., 2026) or to derivatives of the velocity field (Xing et al., 2026). We use the empirical covariance of refined samples and ask whether it predicts error, helps geometric downstream tasks, and can be calibrated post hoc with split conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2023).

Data-dependent source distributions. Flow matching (Lipman et al., 2023) and rectified flow (Liu et al., 2023) learn a vector field that transports a source distribution to the data, and the source need not be fixed noise. PriorGrad (Lee et al., 2022) replaces the standard Gaussian with a data-dependent Gaussian whose mean and variance come from the conditioning signal, and Albergo et al. (2024) formalize transport under such data-dependent couplings. Informed sources have also been used for depth, where transport starts from image latents (Gui et al., 2025), and for robotic control with observation-dependent priors (Machado et al., 2026). FlowMatcher builds its source from the coarse matcher: a per-pixel Gaussian whose mean is the coarse warp and whose scale is predicted from the coarse matching scores, so sampling starts close to the answer and spreads only where the coarse evidence is ambiguous. Like recent clean-data parameterizations (Li & He, 2026; Xu et al., 2026), we predict clean coordinates and derive the velocity from them.

3 METHOD

FlowMatcher splits dense correspondence into coarse matching and stochastic refinement (Fig. 2). For a pair (I_A, I_B), a frozen coarse matcher predicts a correspondence field and a spatially varying source scale derived from its matching scores. A three-stage cascade transports perturbed coordinates using fine-scale evidence. Repeated runs yield hypotheses that can be averaged, selected by inverse consistency, or used to estimate uncertainty.

3.1 COARSE MATCHING AND SOURCE SCALE

The coarse stage supplies both parts of the source distribution: its warp becomes the source mean, and its matching scores determine the source scale. We use the frozen coarse matcher of RoMaV2 (Edstedt et al., 2026). A multi-view Transformer, with alternating within and cross-image

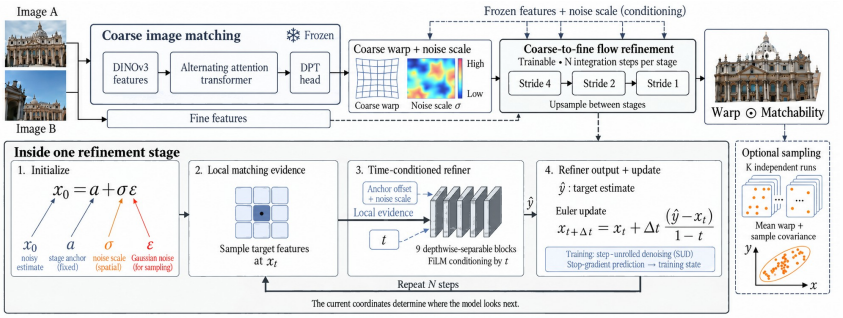


Figure 2: **FlowMatcher architecture.** Top: a frozen coarse matcher predicts a reference warp, and its matching scores are summarized into a spatially varying source scale. Frozen fine features condition a trainable refinement cascade at strides 4, 2, and 1. Bottom: each stage samples candidate coordinates around its reference warp; their current locations determine where local matching evidence is queried, and a time-conditioned refiner predicts clean correspondences that drive N Euler updates.

attention, processes frozen DINOv3 features (Siméoni et al., 2026) and produces token similarities $\mathcal{S}$. For source location p and coarse target token u,

$$\pi_p(u) = \frac{\exp \mathcal{S}_{pu}}{\sum_v \exp \mathcal{S}_{pv}}. \tag{1}$$

A DPT head (Ranftl et al., 2021) decodes from these scores a stride-4 warp $a : \Omega_A \to [-1, 1]^2$ and a matchability $q : \Omega_A \to \mathbb{R}$, where Ω_A is the grid of source pixels. The warp maps each source location to a target location in normalized image coordinates, and the matchability is a logit for whether p has a visible correspondence in the target. Since a is decoded rather than read off as the mean or mode of π_p, the scores carry information that a discards: how concentrated the support is, whether a second mode exists, and whether the decoded match agrees with where the support lies. We summarize this in a descriptor $z(p)$ of six statistics (normalized entropy, the masses of the two strongest modes, spatial dispersion, and the distances from $a(p)$ to the strongest mode and to the score-weighted mean; App. B.3) and map it to a scale

$$\sigma(p) = \exp\big[\mathrm{clip}\big(b + w^\top z(p),\ \ell_{\min},\ \ell_{\max}\big)\big]. \tag{2}$$

The coefficients b, w are fitted once by least squares to log coarse errors on training pairs and then frozen. σ is a proposal width that decides the refinement search radius.

3.2 Refinement

Source and state. We write $W^\star$ for ground truth, W_t for the state at $t \in [0, 1]$, and hats for network outputs. For one stage (omitting the scale index), the conditioning $c = (F_A, F_B, a, \sigma, q)$ collects the source and target feature maps F_A, F_B, the reference warp a, the source scale σ, and matchability q; it is computed once and stays fixed while W_t evolves. We initialize

$$W_0(p) = a(p) + \sigma(p)\,\epsilon(p), \qquad \epsilon(p) \sim \mathcal{N}(0, I_2), \tag{3}$$

independently per pixel; $\epsilon = 0$ gives deterministic inference. The refiner reads the fixed conditioning at the current hypothesis. At each pixel, the lookup

$$\mathcal{E}(W_t; c)(p) = \Big(F_B\big(W_t(p)\big),\ \big\{\langle F_A(p), F_B(W_t(p) + \delta_o)\rangle\big\}_o\Big) \tag{4}$$

samples target features at $W_t(p)$ and correlates them with the source feature over a local window of offsets δ_o (App. B). Only W_t changes between evaluations, so $\mathcal{E}$ is recomputed at every step while c is not. The refiner predicts the clean endpoint and matchability,

$$\big(\widehat{W}_1,\ \widehat{q}\big) = R_\theta\big(\mathcal{E}(W_t; c),\ W_t,\ t;\ c\big), \tag{5}$$

where the direct inputs from c are the source features, matchability, and geometric channels $[(W_t - a)/\sigma_{\mathrm{s}},\ \log(\sigma/\sigma_{\mathrm{s}})]$, with σ_{s} handling resolution change, and W_t also enters through a displacement

embedding. The endpoint is a residual on a, and a sinusoidal time embedding modulates depthwise-separable blocks whose weights are shared across time. Writing $\widehat{W}_1(W_t, t)$ for short, the state follows

$$\frac{dW_t}{dt} = \frac{\widehat{W}_1(W_t, t) - W_t}{1 - t}, \qquad 0 \le t < 1, \tag{6}$$

so each update changes the evidence read by the next prediction.

Relation to flow matching. Conditional flow matching (Lipman et al., 2023; Liu et al., 2023) regresses a velocity $v_\theta(W_t, t)$ onto $W_1 - W_0$ along the linear path $W_t = (1-t)W_0 + tW_1$. With endpoint prediction, $(\widehat{W}_1 - W_t)/(1-t)$ equals this velocity when $\widehat{W}_1 = W_1$, and Euler integration of Eq. 6 is the standard sampler. FlowMatcher keeps the linear path, endpoint parameterization, and Euler sampler, with three changes. (i) The source is the data-dependent Gaussian of Eq. 3, which lets the coarse matcher control initialization. (ii) $\widehat{W}_1$ is supervised against $W^\star$ with a robust coordinate loss rather than by velocity regression. Since $\hat{v} - v = (\widehat{W}_1 - W_1)/(1-t)$ on the linear path, a squared velocity loss weights endpoint errors by $1/(1-t)^2$ and concentrates training on late times; endpoint supervision weights all t equally, and the robust loss limits the influence of the heavy-tailed errors typical of correspondence (occlusions, wrong coarse matches). (iii) Following step-unrolled denoising (Saxena et al., 2023), intermediate states are built from the model's own $t = 0$ prediction rather than from $W^\star$, so the refiner trains on states close to those it visits at inference. Sample variation comes from the state-dependent lookup $\mathcal{E}$, which makes $\widehat{W}_1$ depend on the hypothesis being tested; Sec. 4 evaluates what this variation measures.

3.3 Training and Inference

Training. For each stage we draw W_0 from Eq. 3, set $t = 0$ with probability 0.25 and otherwise $t \sim \mathcal{U}(0, 1)$, and form $W_t = (1-t)W_0 + t\,g$ from a detached preliminary prediction $g = \widehat{W}_1(W_0, 0)$ made in evaluation mode (Alg. 2). Only the second evaluation, $\widehat{W}_1(W_t, t)$, is differentiated. With validity mask M, the endpoint loss is

$$\mathcal{L}_{\text{coord}} = \frac{\sum_p M(p)\left(\|\widehat{W}_1(p) - W^\star(p)\|_2^2 + \eta^2\right)^{1/4}}{\max\left(1, \sum_p M(p)\right)}, \qquad \eta = 10^{-4}s \text{ at stride } s. \tag{7}$$

Each stage adds a binary cross-entropy matchability term with weight 10^{-2}. Losses are summed over stages; the coarse matcher, feature extractors, and scale map stay frozen.

Inference. We integrate Eq. 6 with N uniform Euler steps at $t_n = n/N$ (Alg. 1). The last step has weight one, so $W_1 = \widehat{W}_1(W_{t_{N-1}}, t_{N-1})$, and $N = 1$ reduces to a single prediction from W_0. Three separately parameterized stages operate at strides 4, 2, and 1; each terminal warp and matchability become the reference a and q of the next stage, which resets time, draws fresh noise, and resizes the coarse scale map to its own grid (App. B.4).

3.4 Sampling, Selection, and Uncertainty

Since the source is stochastic, FlowMatcher produces a set of candidate warps, which serve two purposes: proposals to select from and a measure of how strongly the fine-scale evidence determines each correspondence. Independent runs of the cascade produce K terminal warps $\{W^{(k)}\}_{k=1}^K$, sharing features, coarse predictions, and scale maps. The natural summary, the per-pixel mean $\overline{W}(p) = \frac{1}{K}\sum_k W^{(k)}(p)$, is only appropriate when the samples form a single cluster. Where they split into separated groups, for instance between a repeated structure and its neighbor or across an occlusion boundary (Fig. 3), the mean falls between the groups and can be wrong even when one group is correct. We therefore treat the samples as a set of proposals and select among them with a verifier.

Inverse consistency as a verifier. A correct correspondence $A \to B$ should map back to its source under the reverse correspondence $B \to A$. Let W^{rev} be a deterministic prediction in the reverse

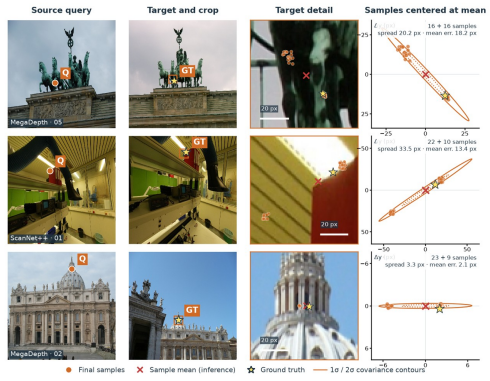


Figure 3: **Sample geometry and correspondence ambiguity.** Each row shows a source query, its ground-truth target, a magnified target region, and the refined samples (orange) centered at their mean (red cross); the star marks ground truth and ellipses the empirical 1σ and 2σ contours. The first two rows show separated hypotheses, where the mean falls between modes; the last shows a concentrated, anisotropic distribution around a well-localized match.

direction, computed once per pair. We score each candidate by its round-trip error and keep the best,

$$r_k(p) = \left\| W^{\mathrm{rev}}\big(W^{(k)}(p)\big) - p \right\|_2, \qquad W^{\mathrm{sel}}(p) = W^{(k^\star(p))}(p), \quad k^\star(p) = \arg\min_k r_k(p), \tag{8}$$

over the K draws and their mean, excluding candidates outside the target image. This technique shows improvement over the mean across all benchmarks. However it remains possible that both directions agree on the same wrong answer, especially on repeated structure (App. F shows failures).

Sample uncertainty. For $K \geq 2$ we summarize disagreement by the per-pixel sample covariance

$$\Sigma(p) = \frac{1}{K-1} \sum_{k=1}^{K} \big(W^{(k)}(p) - \overline{W}(p)\big)\big(W^{(k)}(p) - \overline{W}(p)\big)^\top, \tag{9}$$

which the nonlinear refiner makes anisotropic even though the source is isotropic. We treat Σ as an empirical measure of how strongly the fine-scale evidence determines the result, and test it in Sec. 4.3.

4 Experiments

We evaluate FlowMatcher on dense correspondence and geometric estimation benchmarks, testing its accuracy and the usefulness of its samples for correspondence selection and uncertainty estimation. We also assess robustness to input resolution, the contribution of the heteroscedastic source, and computational cost.

4.1 Dense correspondence

Table 1 covers four held-out populations: outdoor (Li & Snavely, 2018), indoor (Yeshwanth et al., 2023), synthetic (Roberts et al., 2021), and aerial-to-ground (Vuong et al., 2025); App. E.1 adds three more and training-domain diagnostics. The GT oracle picks, the candidate closest to ground truth among the draws and their mean, measuring whether a correct answer is among the proposals. The headroom is large, 5–10 PCK@1 above the sample mean. Without ground truth, the inverse-consistency verifier recovers $+0.52/1.05/0.90/1.13$ PCK@1 over the mean, exceeding RoMaV2 on ScanNet++ v2, Hypersim, and aerial cross-view and trailing it by 1.0 on MegaDepth. We conjecture that further gains are possible with more sophisticated verifiers. FlowMatcher's gap to RoMaV2 is concentrated below one pixel: the deterministic pass trails by 1.1–2.1 points at PCK@1 but by at most 0.3 at PCK@3 and PCK@5, where cycle selection exceeds RoMaV2 on all four benchmarks (six of seven overall, App. E.1). FlowMatcher thus makes fewer medium-sized errors, while RoMaV2 is slightly tighter at the finest scale. We conjecture that training on noisy, self-generated states

Table 1: **Dense correspondence at** 640×640, N=8, K=32. RoMa and UFM use their native inputs (560×560, 420×560). Bold: best non-oracle result per column. The GT oracle picks the lowest-error candidate per pixel and measures headroom only.

	MegaDepth				ScanNet++ v2			
Method	PCK@1↑	PCK@3↑	PCK@5↑	EPE↓	PCK@1↑	PCK@3↑	PCK@5↑	EPE↓
RoMa	75.80	94.31	96.98	2.325	30.12	50.47	58.95	21.562
UFM-Base	52.72	87.14	93.95	2.165	33.91	71.08	83.19	4.721
RoMaV2	**80.24**	95.08	97.32	**1.122**	47.81	79.69	88.97	3.193
Ours det.	78.13	94.90	97.26	1.178	46.54	79.39	88.69	3.246
Ours mean	78.70	94.99	97.30	1.159	46.99	79.59	88.81	3.228
Ours cycle	79.22	**95.15**	**97.41**	1.141	**48.04**	**80.15**	**89.11**	**3.183**
GT oracle	83.78	96.14	97.90	0.932	54.88	82.85	90.56	2.824
	Hypersim				Aerial cross-view			
Method	PCK@1↑	PCK@3↑	PCK@5↑	EPE↓	PCK@1↑	PCK@3↑	PCK@5↑	EPE↓
RoMa	48.95	61.14	65.76	25.322	24.62	58.28	69.93	14.828
UFM-Base	32.57	63.41	73.06	13.204	14.77	47.38	62.95	14.413
RoMaV2	64.46	82.47	88.07	**4.112**	29.91	64.22	77.98	6.343
Ours det.	63.38	82.51	88.03	4.216	28.50	64.57	77.98	6.402
Ours mean	63.85	82.65	88.12	4.180	29.29	65.11	78.25	6.331
Ours cycle	**64.74**	**83.19**	**88.47**	4.150	**30.41**	**66.08**	**78.89**	**6.213**
GT oracle	70.54	85.64	90.07	3.650	39.70	71.49	82.14	5.512

Table 2: **HPatches.** Homography Estimation AUC (%) and AEPE (px) under severity-5 corruptions (Hendrycks & Dietterich, 2019), averaged per family (Table 16).

	Homography AUC↑			AEPE↓				
	@AUC3	@AUC5	@AUC10	Clean	Weath.	Digital	Blur	Noise
RoMaV2	71.82	80.93	89.11	9.95	14.05	8.90	9.89	**8.26**
Ours	**72.40**	**81.42**	**89.38**	**7.71**	**12.83**	**6.59**	**8.25**	8.63

with few Euler steps rewards reaching the correct basin over final sub-pixel precision, which decides one-pixel benchmarks but matters less for robust geometric estimation (Sec. 4.2). Both refiners are trained at 640; at native 1024, the $K=32$ mean exceeds RoMaV2 in PCK@1 on all seven populations (by up to 6.7 on Hypersim), and the deterministic pass exceeds it at PCK@3 and PCK@5 (Table 13). This reflects robustness to a resolution shift: on a common scoring grid, RoMaV2 at 640 remains most accurate (Table 13). On HPatches (Balntas et al., 2017), FlowMatcher likewise makes fewer large errors (Table 2): it improves homography AUC at every threshold and, under weather, digital, and blur corruptions, lowers endpoint error by 9–26%.

4.2 Pose estimation

Table 3 evaluates MegaDepth-1500 (Sun et al., 2021), ScanNet-1500 (Dai et al., 2017; Sarlin et al., 2020), and ETH3D (Schöps et al., 2017). FlowMatcher is best on ETH3D at all thresholds, on par with RoMaV2 on ScanNet, and behind RoMaV2 with its high-resolution pass on MegaDepth. Pose needs few draws: the mean of four is within 0.9 AUC@5 of the mean of 32 everywhere, and cycle selection adds 0.72 and 0.21 AUC@5 over it on MegaDepth and ETH3D (App. E.4). In Table 4, weighting/loss variants within each fixed match-set block share the correspondences and initial RANSAC pose. The reselected rows additionally change the subset used for refinement. The standalone RoMaV2 "RANSAC only" baseline uses its high-resolution pass, whereas its refinement rows start from the standard pass. Refinement minimizes Sampson errors (Larsson & contributors, 2020); uncertainty weighting propagates our sample covariance Σ (Eq. 9) or RoMaV2's predicted covariance to the residual, with parameters cross-fitted on pair halves. (i) *Weighting inliers.* On RANSAC inliers, uncertainty adds 0.64, 0.25, and 2.48 AUC@5 on MegaDepth, ScanNet, and ETH3D, comparable to RoMaV2's head (0.91, 0.19, 2.35). (ii) *Suppressing outliers.* On all 5,000 matches, our uncertainty weighting improves AUC@5 over equal weight-

Table 3: **Relative pose.** Pose AUC (%) at the stated angular thresholds on MegaDepth-1500, ScanNet-1500, and ETH3D. Local evaluations are bidirectional; ours uses $N{=}4$ on MegaDepth/ScanNet and $N{=}8$ on ETH3D. Bold: best per block. Baseline settings in App. E.4.

	MegaDepth			ScanNet			ETH3D		
Method	@5° ↑	@10° ↑	@20° ↑	@5° ↑	@10° ↑	@20° ↑	@5° ↑	@10° ↑	@15° ↑
MASt3R	42.40	61.50	76.90	33.60	**56.80**	**74.10**	79.16	85.37	87.86
LoFTR	52.80	69.20	81.20	22.10	40.80	57.60	62.98	68.68	71.33
DKM	60.40	74.90	85.10	29.40	50.70	68.30	77.90	82.36	84.37
SP+LightGlue	51.00	68.10	80.70	17.80	33.77	52.00	57.11	65.30	69.20
RoMa	62.50	76.60	86.30	31.80	53.40	70.90	82.98	88.40	90.26
UFM Refine	41.08	57.83	72.66	31.77	54.45	72.26	73.27	82.64	86.61
RoMaV2 + HR	**62.56**	**76.66**	**86.39**	34.00	56.52	73.94	82.21	88.38	90.92
Ours ($K{=}4$)	60.12	75.01	85.40	**34.12**	56.61	73.81	**83.91**	**89.46**	**91.70**

Table 4: **Uncertainty in pose refinement.** AUC@5 (%) at 1024. Rows vary the matches and weights of the final refinement (ours: Σ, $K{=}32$; RoMaV2: precision head). RANSAC only: RoMaV2 with its high-resolution pass; refinements start from its standard pass. Bold: better per row.

		MegaDepth		ScanNet		ETH3D	
Matches	Weights / loss	RoMaV2	Ours	RoMaV2	Ours	RoMaV2	Ours
RANSAC only		**62.56**	60.36	**34.00**	33.17	82.21	**83.86**
RANSAC inliers	equal, squared	**65.04**	64.80	34.37	**34.59**	80.20	**83.86**
	uncertainty, squared	**65.95**	65.44	34.56	**34.84**	82.55	**86.34**
Reselected (count-matched)	Sampson residual	67.99	**68.64**	**35.65**	35.28	81.51	**84.98**
	calibrated uncertainty	67.81	**68.37**	**35.68**	35.27	81.52	**85.30**
All 5,000	equal, squared	55.40	**57.17**	**34.15**	33.46	63.47	**64.53**
	uncertainty, squared	61.55	**68.20**	34.94	**35.35**	71.53	**78.77**
	equal, Student-t	64.83	**66.17**	**35.02**	34.79	71.17	**73.28**
	uncertainty, Student-t	67.49	**69.19**	**35.37**	35.19	77.25	**82.53**

ing by $11.0/1.9/14.2$ points with squared loss ($6.2/0.8/8.1$ for RoMaV2) and $3.0/0.4/9.3$ with Student-t loss ($2.7/0.4/6.1$), in MegaDepth/ScanNet/ETH3D order. Our weighted Student-t refinement achieves the best MegaDepth score in Table 4 (69.19). (iii) *Setting an inlier threshold.* Calibrated-uncertainty re-selection (Sec. 4.3) adds 2.9 AUC@5 on MegaDepth over uncertainty-weighted RANSAC-inlier refinement (1.9 for RoMaV2), but loses about 1.0 on ETH3D for both models. Count-matched Sampson selection performs similarly: calibration supplies a threshold, without demonstrating a consistent ranking advantage.

4.3 Uncertainty

We use uncertainty for three purposes: ranking matches, weighting them (Sec. 4.2), and bounding their errors. Following Ilg et al. (2018), we discard pixels from most to least uncertain and track the error of the remaining ones; AUSE is the area between this curve and the curve for the ideal ordering by true error (Table 5; curves in Fig. 4). As a learned baseline, we train a precision head in the manner of RoMaV2 jointly with our refiners (App. C.3), so both signals rank the errors of the same prediction. The spread of 32 draws ranks errors better than this head on ScanNet++ v2 and Hypersim and ties it on MegaDepth; four draws already improve EPE-AUSE on both indoor populations populations. Neither matchability nor σ alone ranks well (Sec. 4.4). Thresholds, inlier tests, and confidence maps such as $P(\|e\| < R)$ (Truong et al., 2021) need more than an ordering: they need the uncertainty in pixels, including how often large errors occur. The raw spread is too narrow: its nominal 90% regions cover only 12–27% of errors, because it measures how much the result depends on the starting point rather than the full error distribution. We therefore calibrate it on held-out pairs (Kuleshov et al., 2018). Rescaling by a single split-conformal factor (Vovk et al., 2005; Angelopoulos & Bates, 2023) gives 90.25–90.52% held-out coverage on every population and inference setting we tested (App. E.5). Because errors are heavy-tailed, however, a factor fitted at

Table 5: **Uncertainty** at 1024, $N{=}8$. Left: EPE AUSE when ranking by matchability, σ, Trained precision head ($\mathrm{tr}\, P^{-1}$), or the spread of K draws. Right: held-out coverage (%) of nominal 90% regions, raw and split-conformal, and Student-t calibration error (pp) over levels 50–99%.

	EPE AUSE ↓					Coverage@90		
Population	Overlap	σ only	Trained	$K{=}4$	$K{=}32$	Raw	Conformal	t-MAE ↓
MegaDepth	0.251	0.430	0.203	0.243	**0.202**	19.1	90.4	0.64
ScanNet++ v2	0.442	0.452	0.263	0.247	**0.204**	12.5	90.4	0.95
Hypersim	0.295	0.473	0.233	0.215	**0.171**	26.8	90.3	0.20

Table 6: **Source ablation** on MegaDepth. The homoscedastic control is trained on MegaDepth only.

Source (steps)	PCK@1↑	EPE↓	AUSE↓
Homosced. (100k)	76.20	1.241	**0.195**
Heterosced. (100k)	**77.67**	**1.191**	0.200

Table 7: **Cost** at 640, A100, batch one: latency (ms) and peak memory (GiB).

Setting	Latency	Memory
RoMaV2	94.2 (1.0×)	3.20
Ours, $N{=}4$, $K{=}1$	245.6 (2.6×)	1.82
Ours, $N{=}4$, $K{=}4$	851 (9.0×)	1.82

90% misses other levels by 9–13 points. A bivariate Student-t model (Lange et al., 1989) with scale matrix $\tau^2\Sigma + \lambda^2 I$ and 1–2.4 degrees of freedom, fitted by maximum likelihood, achieves calibration MAE below 1 percentage point in Table 5 and below 2 points across all tested settings. Its per-pixel probability of a sub-pixel match is accurate to 3–4 points on MegaDepth, and its calibrated scale sets the inlier threshold used in Sec. 4.2. Calibration must be refitted per domain and setting (App. E.5).

4.4 Heteroscedastic versus homoscedastic source

Table 6 compares FlowMatcher trained only on MegaDepth for 100K steps with a homoscedastic variant trained under the same conditions, drawing every source sample with one constant scale. We observe that the heteroscedastic source is more accurate (77.67 vs. 76.20 PCK@1, 1.191 vs. 1.241 px EPE). We attribute this to adapting the search radius to the coarse evidence. Both rank errors about equally well, and σ alone ranks far worse than four draws (Table 5), since it is constant within a coarse token and predicts the coarse rather than the refined error. This suggests that the final ranking comes from how much refined hypotheses disagree, not from the source alone.

4.5 Computational cost

Cost scales with the number of refiner evaluations, $3NK$ per pair; features and coarse matches are computed once and shared (Table 7; full grid in App. E.6). A single $N{=}4$ draw costs 2.6× RoMaV2's latency with under 60% of its memory and reaches 78.11 PCK@1 on MegaDepth (78.13 for the $N{=}8$ deterministic pass). Four $N{=}4$ draws cost 9.0× without additional memory and already suffice downstream: they match 32 draws in pose (Sec. 4.2). Draws parallelize across devices.

5 Conclusion

FlowMatcher turns the ambiguity exposed by coarse matching into a stochastic refinement whose samples serve two purposes. As proposals, they contain matches the mean misses, partly recovered by an inverse-consistency verifier. As an uncertainty signal, their spread is competitive with the trained head in EPE-AUSE, improves pose as weights or an inlier test, and is calibrated by a post-hoc fit. The heteroscedastic source improves accuracy; the uncertainty comes from the transport. The main limitation is compute: each draw is a full refiner pass, so cost grows linearly with K, and four draws at $N{=}4$ cost 9× RoMaV2 (Sec. 4.5). The right number of draws depends on the task. Pose is competitive with four, whereas dense correspondence benefits from more, in accuracy (up to +1.5 PCK@1 from one to 32 draws at 1024; App. E.3), error ranking (Sec. 4.3). Beyond compute, all samples start from the same frozen coarse warp and can share an error that their spread cannot reveal. Allocating draws to uncertain pixels, sharing computation across draws, distillation, and stronger verifiers could lower this cost and close the gap to the oracle.

AI USE STATEMENT

We used large language model assistants to help write and debug experiment code and evaluation scripts, and to edit the manuscript for clarity. They were not used to generate research ideas or results. All code was reviewed and tested by the authors, and every reported number comes from runs executed by the authors. The authors take full responsibility for the content of this paper.

ETHICS STATEMENT

This work uses publicly available image-matching datasets and benchmarks and involves no human subjects. Dense correspondence supports 3D reconstruction and localization, including of places where people appear in photographs; we do not introduce new data collection or identity-related capabilities.

REPRODUCIBILITY STATEMENT

App. B specifies the architecture, the source-scale statistics and fitted coefficients, and the inference procedure (Alg. 1). App. C specifies the training data, labels, objective (Alg. 2), and optimization schedule. App. D defines every metric and evaluation protocol. We will release code and weights.

REFERENCES

Michael S. Albergo, Mark Goldstein, Nicholas M. Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. In *International Conference on Machine Learning*, 2024.

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *European Conference on Computer Vision*, 2022.

Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020. URL https://arxiv.org/abs/2001.10773.

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*, 2017.

Friso de Kruiff, Dario Coscia, Max Welling, and Erik J. Bekkers. Flowing with confidence. *arXiv preprint arXiv:2605.18472*, 2026. URL https://arxiv.org/abs/2605.18472.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17765–17775, 2023.

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19790–19800, 2024.

Johan Edstedt, David Nordström, 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*, 2026.

Ming Gui, Johannes Schusterbauer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. DepthFM: Fast generative monocular depth estimation with flow matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3203–3211, 2025. doi: 10.1609/aaai.v39i3.32330.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Eddy Ilg, Özgün Çiçek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *European Conference on Computer Vision*, 2018.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9492–9502, 2024.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Kenneth L. Lange, Roderick J. A. Little, and Jeremy M. G. Taylor. Robust statistical modeling using the t distribution. *Journal of the American Statistical Association*, 84(408):881–896, 1989.

Viktor Larsson and contributors. PoseLib: Minimal solvers for camera pose estimation. `https://github.com/vlarsson/PoseLib`, 2020.

Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. PriorGrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *International Conference on Learning Representations*, 2022.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision*, 2024.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

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*, pp. 2041–2050, 2018.

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*, pp. 17627–17638, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://arxiv.org/abs/1711.05101`.

Ao Luo, Xin Li, Fan Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. FlowDiffuser: Advancing optical flow estimation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19167–19176, 2024.

Ao Luo, Xin Li, Fan Yang, Yuezun Li, Zhaoquan Yuan, Shan Zhao, Bing Su, and Xiao Wu. Optical flow matching: Reframing optical flow as continuous transport dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28669–28678, 2026.

Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual–inertial localization revisited. *The International Journal of Robotics Research*, 39(9):1061–1084, 2020. doi: 10.1177/0278364920931151.

Bruno Machado, Alexandre Chapin, Emmanuel Dellandréa, and Liming Chen. Flowing with purpose: Latent action guided flow matching policies for robotic manipulation. *arXiv preprint arXiv:2606.23420*, 2026. URL https://arxiv.org/abs/2606.23420.

Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4040–4048, 2016.

Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. doi: 10.1109/TRO.2015.2463671.

Jisu Nam, Gyuseong Lee, Sunwoo Kim, Hyeonsu Kim, Hyoungwon Cho, Seyeon Kim, and Seungryong Kim. Diffusion model for dense matching. In *International Conference on Learning Representations*, 2024.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. URL https://arxiv.org/abs/1709.07871.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10912–10922, 2021.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019.

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*, 2020.

Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J. Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. In *Advances in Neural Information Processing Systems*, 2023.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.

Johannes L. Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, 2016.

Thomas Schöps, Johannes L. Schönberger, 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*, 2017.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1409.1556.

Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008. doi: 10.1007/s11263-007-0107-3.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. URL https://arxiv.org/abs/2104.00680.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, 2020.

Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. SMD-Nets: Stereo mixture density networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8942–8952, 2021.

Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. PDC-Net+: Enhanced probabilistic dense correspondence network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10247–10266, 2023. doi: 10.1109/TPAMI.2023.3249225.

Matthieu Vilain, Rémi Giraud, Yannick Berthoumieu, and Guillaume Bourmaud. Handling multiple hypotheses in coarse-to-fine dense image matching. In *2025 IEEE International Conference on Image Processing (ICIP)*, pp. 1265–1270, 2025. doi: 10.1109/ICIP55913.2025.11084489.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Khiem Vuong, Anurag Ghosh, Deva Ramanan, Srinivasa Narasimhan, and Shubham Tulsiani. AerialMegaDepth: Learning aerial-ground reconstruction and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

Qianliang Wu, Haobo Jiang, Lei Luo, Jun Li, Yaqing Ding, Jin Xie, and Jian Yang. Diff-Reg: Diffusion model in doubly stochastic matrix space for registration problem. In *European Conference on Computer Vision*, 2024.

Jiarui Xing, Song Wang, and Jian Wang. Divergence is uncertainty: A closed-form posterior covariance for flow matching. *arXiv preprint arXiv:2605.00941*, 2026.

Haofei Xu, Rundi Wu, Philipp Henzler, Nikolai Kalischek, Michael Oechsle, Fabian Manhardt, Marc Pollefeys, Andreas Geiger, Federico Tombari, and Michael Niemeyer. PointDiT: Pixel-space diffusion for monocular geometry estimation. In *International Conference on Machine Learning*, 2026.

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*, 2020.

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*, pp. 12–22, 2023.

Shihua Zhang and Jiayi Ma. DiffGlue: Diffusion-aided image feature matching. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024. doi: 10.1145/3664647.3681069.

Yuchen Zhang, Nikhil Keetha, Chenwei Lyu, Bhuvan Jhamb, Yutian Chen, Yuheng Qiu, Jay Karhade, Shreyas Jha, Yaoyu Hu, Deva Ramanan, Sebastian Scherer, and Wenshan Wang. UFM: A simple path towards unified dense correspondence with flow. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Fengyuan Zuo, Haiyan Jin, Yuanlin Zhang, Zhaolin Xiao, Bin Wang, and Yuerong Mu. FlowFM: Advancing dark optical flow estimation with flow matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6837–6846, 2026.

A Overview of the appendix

The appendix is organized to follow the main text. Table 8 maps each main-text claim to its supporting material. App. B and App. C give the complete model and training recipe behind Sec. 3; App. D defines every metric and protocol used in Sec. 4; App. E reports the complete results, controls, and weaker results behind each experiment; and App. F shows qualitative examples, including failure cases. All results use the single final checkpoint of the main text.

Table 8: **Where to find supporting evidence.**

Main-text element	Supporting material
Source scale, Eq. 2	Statistics, fitted coefficients and fit quality (App. B.3)
Refiner and dynamics, Eqs. 5–6	Architecture (App. B), inference (Alg. 1)
Training, Eq. 7	Data, labels, objective (Alg. 2), schedule (App. C)
Dense accuracy, Table 1	Three further held-out and two training-domain populations (App. E.1)
Resolution robustness, Sec. 4.1	Native-1024 results on all populations, common-grid scoring (App. E.2)
HPatches, Table 2	Individual corruptions, illumination and viewpoint subsets (App. E.4)
Pose, Tables 3–4	Pose for every K, cycle-selection intervals, baseline settings (Apps. E.3 and E.4)
Uncertainty, Table 5	Ranking intervals and intermediate K; calibration across populations and settings (App. E.5)
Trained precision head, Sec. 4.3	Architecture (App. B) and loss (App. C.3)
Source ablation, Table 6	Flat-source and scale-channel interventions at inference (App. E.3)
Cost, Table 7	$N \times K$ accuracy grids (App. E.3); latency and memory at three resolutions (App. E.6)

B Architecture

B.1 Components and coordinates

Only the three refiners are trained. The frozen modules are those of the original RoMaV2 release (Edstedt et al., 2026): the DINOv3 ViT-L/16 descriptor (Siméoni et al., 2026), the coarse matcher, and the VGG19-BN fine-feature pyramid (Simonyan & Zisserman, 2015). The descriptor supplies concatenated layer-11 and layer-17 features to the coarse matcher, a multi-view transformer with a DPT head that returns a stride-four warp and an overlap logit. Its token similarities, divided by temperature 0.1 and normalized over target tokens, give the distribution π_p of Eq. 1. At input 640 the token grid is 40×40, i.e., 1600 target candidates per source token. VGG features are taken before the first three max-pooling layers. The frozen modules run in evaluation mode. We replace RoMaV2's refiners with ours; its original refiners and high-resolution second pass are not used by FlowMatcher.

A warp stores normalized target coordinates: target column j has coordinate $2(j + 1/2)/W_B - 1$, and rows use H_B analogously. With $T_B = \mathrm{diag}(W_B/2, H_B/2)$, a normalized residual r is $T_B r$ in pixels, and a covariance transforms as $S_{\mathrm{px}} = T_B S_{\mathrm{norm}} T_B^\top$. Interpolation uses the pixel-center convention. States and predictions are not clipped to the image. Table 9 summarizes the refiners.

B.2 Evidence lookup and conditioning

At stage ℓ, the reference warp a_ℓ is the resized output of the preceding stage (the coarse warp at stride four) and stays fixed within the stage; the state x evolves. A shared linear projection maps both images' fine features to d_f channels, and target features are sampled bilinearly at x. On a grid of width W_ℓ and height H_ℓ, the correlation for integer window offset o is

$$L_\ell(p, o; x) = \frac{f_A(p)^\top f_B(x(p) + \delta_{\ell,o})}{\sqrt{d_f}}, \qquad \delta_{\ell,o} = \mathrm{diag}(2/W_\ell, 2/H_\ell) F_R o, \tag{10}$$

with resolution factor $F_R = \mathrm{diag}(W_A/640, H_A/640)$. The displacement embedding receives the scaled normalized displacement $F_R(x - p)$, and a softmax-weighted mean window offset adds two channels. Source features, sampled target features, displacement embedding, correlations, and the

Table 9: **Refinement architecture.** Grids and pixel equivalents refer to a 640×640 input. Each stage has its own weights, shared across its integration steps. Parameter counts exclude BatchNorm running statistics.

	Stride 4	Stride 2	Stride 1
Grid	160×160	320×320	640×640
Feature channels $\to$ projection	$256 \to 192$	$128 \to 48$	$64 \to 12$
Displacement embedding	79	23	8
Correlation window	7×7	3×3	3×3
Correlation + offset channels	$49 + 2$	$9 + 2$	$9 + 2$
Trunk width	514	130	43
Depthwise blocks / kernel	9 / 5×5	9 / 5×5	9 / 5×5
Coordinate-head scale λ_ℓ	0.2	0.1	0.05
Stage scale α_ℓ	0.019	0.003	0.0016
Source-scale range, pixels	1.52–48.64	0.24–7.68	0.128–4.096
Trainable parameters	3,793,177	528,481	161,974

offset cue are concatenated. A zero-initialized 1×1 projection adds the geometric conditioning channels specified after Eq. 5. Each of nine blocks applies a depthwise 5×5 convolution, BatchNorm (momentum 0.01), time-conditioned affine modulation (Perez et al., 2018), ReLU, and a pointwise convolution, without residual connections. The time embedding concatenates sines and cosines of $1000t$ at 64 geometrically spaced frequencies, followed by a width-128 Linear–SiLU–Linear network; each block has its own modulation layer.

Prediction heads. Here $R_{\theta,\ell}(x,t)$ denotes the coordinate component of the refiner output in Eq. 5. A zero-initialized two-channel head predicts

$$R_{\theta,\ell}(x,t) = a_\ell + \lambda_\ell h_{\theta,\ell}(x,t). \tag{11}$$

Each stage also outputs a matchability logit, supervised by binary cross-entropy against covisibility and passed to the next stage. An auxiliary precision head follows RoMaV2 (Edstedt et al., 2026): three outputs z_{11}, z_{21}, z_{22} define $P = LL^\top$, where L is lower triangular with $l_{11} = \mathrm{Softplus}(z_{11}) + 10^{-6}$, $l_{21} = z_{21}$, and $l_{22} = \mathrm{Softplus}(z_{22}) + 10^{-6}$. This precision output is used only for the learned uncertainty baseline in Sec. 4.3, not for FlowMatcher's sampling, cycle selection, or pose refinement.

Relative to the released RoMaV2 refiners, our stages add a stride-one correlation window, the offset cue, source conditioning, and time modulation; trunk widths change from $512/128/32$ to $514/130/43$. The three refiners have 4,483,632 trainable parameters.

B.3 Source scale

For target-token probabilities π_j and normalized token positions z_j, let $\mu = \sum_j \pi_j z_j$ and $v^2 = \sum_j \pi_j \|z_j\|^2 - \|\mu\|^2$. We extract modes greedily: starting at the largest remaining weight, collect a disk of radius two tokens, suppress it, and repeat. Let m_1, m_2 be the first two disk masses, b_1 the weighted center of the first disk, and a_c the coarse warp averaged over the source token. The six statistics are

$$S_{\mathrm{feat}} = \left[H,\ 1 - m_1,\ m_2/(m_1 + 10^{-8}),\ v,\ \|b_1 - a_c\|,\ \|\mu - a_c\|\right],$$

with normalized entropy $H = -\sum_j \pi_j \log \pi_j / \log n_c$. The fitted map and its per-stage version are

$$\begin{aligned} \widetilde{\sigma} &= \exp\big(\mathrm{clip}(\beta^\top S_{\mathrm{feat}} + b, -8, -1.2)\big), \\ \sigma_\ell &= \mathrm{clip}(\mathcal{U}_\ell \widetilde{\sigma},\ 0.25\alpha_\ell,\ 8\alpha_\ell), \end{aligned} \tag{12}$$

with bilinear resizing $\mathcal{U}_\ell$ and

$$\beta = (5.97, -3.76, 3.81, -1.42, 1.48, 1.05)^\top, \quad b = -5.49.$$

Fitting. Before refiner training, ordinary least squares fits the log coarse error $\log e$, with e clipped to $[10^{-4}, 0.5]$, to S_{feat}, where e is the mean radial error over valid pixels of a coarse token in normalized coordinates. The fit uses 640-input training pairs from MegaDepth, ScanNet++v2, Hypersim, and Map-Free, keeping 213,765 tokens with valid fraction above 0.25. The fit is deliberately simple: σ only has to set a plausible search width, and it is frozen during refiner training and inference.

B.4 Inference

Algorithm 1 gives one draw of the cascade. With N steps per stage and $t_k = k/N$,

$$x_{k+1} = x_k + \frac{1/N}{\max(1 - t_k, 10^{-3})} \big[R_{\theta,\ell}(x_k, t_k) - x_k\big]; \tag{13}$$

the guard is inactive for the evaluated step counts, and the final weight is one. One draw makes $3N$ refiner calls. Frozen features and coarse outputs are shared across the K draws. Deterministic inference sets $\epsilon = 0$ at every stage and need not equal the sample mean. Cycle selection (Eq.8) uses a deterministic reverse prediction, excludes out-of-image candidates, and falls back to the mean when no candidate is valid.

Algorithm 1 FlowMatcher inference (one draw)

Require: coarse warp a, matchability q, coarse scale map $\widetilde{\sigma}$, features, steps N
1: **for** stage ℓ at strides $4, 2, 1$ **do**
2: $a_\ell, q_\ell \leftarrow$ resize previous output (or a, q at stride 4)
3: $\sigma_\ell \leftarrow \text{clip}(\mathcal{U}_\ell \widetilde{\sigma},\ 0.25\alpha_\ell,\ 8\alpha_\ell)$
4: $x_0 \leftarrow a_\ell + \sigma_\ell\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$ ($\epsilon = 0$ for deterministic inference)
5: **for** $k = 0, \ldots, N-1$ **do**
6: $t_k \leftarrow k/N$
7: $x_{k+1} \leftarrow x_k + \frac{1/N}{1-t_k}\big[R_{\theta,\ell}(x_k, t_k) - x_k\big]$ (Eq.13)
8: **end for**
9: pass x_N and updated confidence to the next stage
10: **end for**
11: **return** x_N at stride 1

C Training

The refiners are trained from scratch on top of the frozen released modules; the coarse matcher is not retrained.

C.1 Data mixture

Each training example first picks a dataset with probability proportional to its weight (Table 10; weights sum to 7.52, so a weight-one dataset receives 13.3% of draws) and then a pair uniformly within it, except for the scene weighting of MegaDepth described below. Images are resized to 640×640 without preserving aspect ratio, intrinsics are rescaled per axis, and depth is resized by nearest neighbor.

MegaDepth and AerialMegaDepth. MegaDepth uses the LoFTR scene files (Sun et al., 2021), and equivalent AerialMegaDepth files are built from its COLMAP models. Both exclude the evaluation scenes 0015 and 0022. Per scene, up to 100,000 pairs with overlap above 0.01 and 100,000 above 0.35 are concatenated, and a scene with n_s entries is sampled proportionally to $n_s^{0.75}$. Aerial pairs require at least 16 shared tracks and overlap above 0.05, keeping at most 30,000 highest-overlap pairs per scene.

BlendedMVS and Hypersim. BlendedMVS uses the 768×576 release; one malformed scene is excluded. Ordered pairs require depth overlap of at least 0.05, with at most 1,500 per scene. Hypersim uses tone-mapped images from scenes with volume index below 50; indices 50 and above are reserved for evaluation. Euclidean ray distance is converted to planar depth with the scene projection matrix. Within-trajectory pairs require overlap of at least 0.2, with at most 300 per scene.

Table 10: **Refiner training mixture.** Counts are indexed pair entries, not unique images. Classic ScanNet has weight zero.

Dataset	Label source	Weight	Scenes	Pair entries
MegaDepth (Li & Snavely, 2018)	MVS depth	1	169	9,903,321
AerialMegaDepth (Vuong et al., 2025)	MVS depth	1	124	4,180,296
BlendedMVS (Yao et al., 2020)	Mesh depth	1	388	565,757
Hypersim (Roberts et al., 2021)	Rendered depth	1	399	118,987
TartanAir-WB (Wang et al., 2020)	Rendered flow	1	5	12,001
ScanNet++ v2 (Yeshwanth et al., 2023)	Mesh depth	1	853	7,920,834
Map-Free (Arnold et al., 2022)	Reconstructed MVS depth	1	429	75,019
FlyingThings3D (Mayer et al., 2016)	Rendered flow	0.5	2,239	161,208
Virtual KITTI 2 (Cabon et al., 2020)	Rendered flow	0.01	5	84,840
UnrealStereo4K (Tosi et al., 2021)	Rendered disparity	0.01	8	16,000

ScanNet++ and Map-Free. ScanNet++ uses DSLR images from the `nvs_sem_train` split with depth rendered from the laser-scan mesh; pairs require bidirectional overlap of at least 0.2, with at most 10,000 per scene. Map-Free supplies poses but no depth, so we reconstruct depth with COLMAP MVS (Schönberger et al., 2016) from the supplied poses, using at most 120 frames per scene, and cap pairs at 200 per scene.

Flow and stereo data. FlyingThings3D uses the official training split (frames 6–15, both cameras, both directions, clean and final passes). Virtual KITTI 2 uses all scenes, rendering variants, and cameras with consecutive frames. UnrealStereo4K uses the quarter-resolution release with scenes 00000–00007 for training and 00008 held out. The TartanAir member is UFM's wide-baseline store (Zhang et al., 2025).

Held-out evaluation data. MegaDepth scenes 0015 and 0022 (also used for AerialMegaDepth), ScanNet++ validation scenes, Hypersim volumes 50 and above, FlyingThings3D test sequences, and UnrealStereo4K scene 00008 are excluded from refiner training.

C.2 Labels and augmentation

For depth-labeled data, labels are recomputed after augmentation at each stage's grid: source pixel centers are back-projected, transformed into the target camera, and projected. A pixel is covisible if both depths are positive, the projection lies inside the target image, and the bilinearly sampled target depth d_B satisfies $|z - d_B|/d_B < 0.05$. For flow and disparity data, the native field gives target coordinates, and visibility combines image bounds, supplied masks, and a forward–backward check (1.5 px; 1 px for UnrealStereo4K). Invalid coordinates are excluded from the coordinate loss.

Horizontal flips (probability 0.5) are shared by the pair, with depths, intrinsics, and flow labels transformed consistently. MegaDepth and AerialMegaDepth additionally receive a shared integer translation in $[-32, 32]^2$. Per image, photometric augmentation applies grayscale with probability 0.1, brightness scaling uniform in $[1/1.5, 1.5]$, and hue shifts uniform in $[-15°, 15°]$.

C.3 Objective

Algorithm 2 gives one training step for one stage and direction. The preliminary pass runs with current parameters in evaluation mode and without gradients, leaving BatchNorm statistics unchanged, following step-unrolled denoising (Saxena et al., 2023). Supervision compares $\widehat{y}_\ell$ with the ground-truth warp y_ℓ, not with g. Detached outputs of each stage become the next stage's reference, so losses do not propagate across stages. With covisibility mask M_ℓ, stride s_ℓ, residual $e = \widehat{y}_\ell - y_\ell$, and $\eta_\ell = 10^{-4} s_\ell$,

$$\mathcal{L}_\ell = \mathcal{L}_{\text{coord}} + 10^{-2}\mathcal{L}_{\text{overlap}},$$
$$\mathcal{L}_{\text{coord}} = \frac{\sum_p M_\ell(p)(\eta_\ell^2 + \|e(p)\|^2)^{1/4}}{\max(1, \sum_p M_\ell(p))}. \tag{14}$$

Table 11: **Training and optimization settings.**

Setting	Value
Trainable parameters	Three refiners, 4.48M; descriptor, coarse matcher and VGG features frozen
Batch per optimizer step	64 pairs: 2/GPU $\times$ 2 accumulated microbatches $\times$ 16 A100 GPUs
Directions	Both directions for every pair; losses averaged
Optimizer	AdamW, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01
Learning rate	Peak 4×10^{-4}; 1,000-step linear warmup; cosine schedule
Gradient clipping	Global norm 1; non-finite gradients skip the update
Precision	BF16 autocast; FP32 weights, optimizer state and source statistics
Steps / seed	300,000 / 0

Each stage also trains the precision head of App. B, as in RoMaV2, with the Gaussian negative log-likelihood of the detached residual, $\mathcal{L}_{\text{prec}} = \frac{1}{2} e^\top P e - \frac{1}{2} \log\det P$, on covisible pixels with $\|e\| < 8$ px; detaching e keeps this term from affecting the predicted warp. For clarity, it is omitted from Eq. 14 and Alg. 2. The three stage objectives have equal weight, and the two matching directions are averaged.

Algorithm 2 FlowMatcher training step (one stage ℓ, one direction)

Require: reference a_ℓ, scale σ_ℓ, conditioning c_ℓ, ground truth y_ℓ, mask M_ℓ

$X_0 \leftarrow a_\ell + \sigma_\ell\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$

$t \leftarrow 0$ with probability 0.25, otherwise $t \sim \mathcal{U}(0, 1)$

$g \leftarrow \text{stopgrad}\big(R_{\theta,\ell}(X_0, 0)\big)$ (evaluation mode, no gradient)

$X_t \leftarrow (1 - t)X_0 + t\,g$

$\widehat{y}_\ell \leftarrow R_{\theta,\ell}(X_t, t)$

$\mathcal{L}_\ell \leftarrow \mathcal{L}_{\text{coord}}(\widehat{y}_\ell, y_\ell; M_\ell) + 10^{-2}\mathcal{L}_{\text{overlap}}$ (Eq.14)

pass $\text{stopgrad}(\widehat{y}_\ell)$ and confidence to stage $\ell + 1$ as reference

return $\mathcal{L}_\ell$

C.4 Optimization

Table 11 lists the optimization settings. We use AdamW (Loshchilov & Hutter, 2019), and each resume restores weights and optimizer moments. Training took approximately 4,270 A100 GPU-hours.

D Metrics and protocols

Input resolution and scoring grid are separate choices: a one-pixel tolerance at 1024 is stricter, in normalized coordinates, than at 640. Accuracy and coverage are percentages and differences between them are percentage points (pp). Lower is better for error, AUSE, calibration MAE, and latency; higher is better for PCK, MMA, AUC, and success.

D.1 Dense correspondence

For predicted and ground-truth target coordinates $\widehat{w}_i(p)$ and $w_i^*(p)$ of source pixel p in pair i, on a scoring grid with target size (W_i, H_i),

$$e_i(p) = \big\|\text{diag}(W_i/2, H_i/2)\big[\widehat{w}_i(p) - w_i^*(p)\big]\big\|_2,$$
$$\text{EPE} = \frac{\sum_i \sum_{p \in V_i} e_i(p)}{\sum_i |V_i|}, \qquad \text{PCK@}d = 100\,\frac{\sum_i \sum_{p \in V_i} \mathbf{1}[e_i(p) < d]}{\sum_i |V_i|}, \tag{15}$$

pooling all ground-truth-valid pixels V_i without confidence filtering. Common-grid comparisons (Table 13) transport predictions to the 640 grid before scoring.

HPatches corruptions. AEPE averages endpoint error over native-resolution query points from the known homography within each corruption condition. Stochastic corruptions use five seeds; the mean is over the 13 appearance corruptions of Hendrycks & Dietterich (2019), excluding clean images and the geometry-changing glass-blur and elastic transformations.

D.2 Relative pose and homography

For estimated and true rotations R, R^* and translation directions t, t^*, the pose error in degrees is $e_{\text{pose}} = \max(e_R, e_t)$ with

$$e_R = \arccos\big(\text{clip}((\text{tr}(R^{*\top}R) - 1)/2, -1, 1)\big), \qquad e_t = \arccos\big(\text{clip}(|t^\top t^*|/(\|t\|\|t^*\|), 0, 1)\big). \tag{16}$$

Failed estimates count as $90°$. AUC@T is the normalized area under the recall curve up to T, computed by the trapezoidal rule over sorted errors with $(0, 0)$ prepended and (T, r_{last}) appended. ETH3D (Schöps et al., 2017) evaluation uses 4,903 pairs from 13 scenes, the original cameras, PoseLib (Larsson & contributors, 2020) with an epipolar threshold of one pixel at 640 long edge, and five estimator seeds.

HPatches homography. HPatches (Balntas et al., 2017) homography AUC@3/5/10 uses the mean corner error per pair on all 116 sequences and 580 pairs, and five trials.

D.3 Uncertainty ranking

For K draws at a pixel, $\bar{w}$ and S are the sample mean and the unbiased sample covariance (Eq. 9); S describes variability among model draws, not correspondence error against ground truth. Spread ranking uses $\text{tr}\, S$, the trained precision head uses $\text{tr}(P^{-1})$, and overlap ranks higher overlap logits as more reliable. All signals rank the same deterministic prediction. For removal fraction $q \in \{0, 0.02, \ldots, 1\}$ we keep the $\lceil (1-q)n \rceil$ least uncertain of at most 4,096 valid pixels per pair (the same pixels for every signal) and average retained errors over pairs. With retention curve $c(q)$ and oracle curve $c^*(q)$ obtained by sorting true errors,

$$\text{AUSE} = \left| \text{trapz}_q \frac{c(q)}{z} - \text{trapz}_q \frac{c^*(q)}{z} \right|, \qquad z = \max_q c^*(q) + 10^{-6}, \tag{17}$$

averaged over pairs. AUSE measures ranking relative to each prediction's own oracle, not accuracy.

D.4 Coverage and calibration

Coverage evaluates regions around $\bar{w}$ in pixels. The raw reference region for $K > 2$ is $(\bar{w} - w^*)^\top S^{-1}(\bar{w} - w^*) \le \frac{(K+1)(K-1)}{K(K-2)}\, 2F^{-1}_{2,K-2}(a)$ at requested level a, the predictive region that would hold if ground truth were an additional independent draw; its empirical coverage tests exactly that assumption. The *conformal* scalar is the $\lceil (n_{\text{cal}} + 1)0.9 \rceil / n_{\text{cal}}$ quantile of the squared Mahalanobis residual on a random half of the pairs, applied to the other half (Vovk et al., 2005; Angelopoulos & Bates, 2023). The *Student-t* model fits a bivariate t distribution with scale matrix $A = \tau^2 S + \lambda^2 I$ and ν degrees of freedom by likelihood on the calibration half. Coverage averages pixels within a pair and then pairs, and $\text{MAE}_{\text{cal}} = \frac{100}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} |\widehat{C}(a) - a|$ over $\mathcal{A} = \{.50, .55, \ldots, .95, .99\}$. We use 200 random splits for scalar coverage and 20 for MAE.

E Additional results

This section gives the complete evidence behind Sec. 4 in the order of the main text, including results where FlowMatcher is weaker. Main-text tables are not repeated.

E.1 Dense correspondence at 640

The seven held-out dense populations each contain 500 pairs (400 for UnrealStereo4K): MegaDepth, ScanNet++ v2, Hypersim, aerial mixed and aerial cross-view (from AerialMegaDepth), FlyingThings3D, and UnrealStereo4K. Table 1 reports four of them and Table 12a the other three. Table 12b reports BlendedMVS and Virtual KITTI 2 (500 pairs each, drawn from training scenes).

Table 12: **Additional dense populations at 640.** PCK@d (%) and EPE (px); N=8, K=32 shared draws. (a) The three held-out populations not in Table 1. (b) Populations drawn from training scenes. Bold: best per column.

(a) Held-out

	Aerial mixed				FlyingThings3D				UnrealStereo4K			
Method	@1↑	@3↑	@5↑	EPE↓	@1↑	@3↑	@5↑	EPE↓	@1↑	@3↑	@5↑	EPE↓
UFM-Base	28.87	64.16	75.49	12.000	85.99	95.63	97.36	1.010	81.31	94.28	97.18	0.895
RoMa	55.27	75.85	81.41	13.082	84.77	91.23	93.03	3.455	87.94	93.67	95.69	0.802
RoMaV2	**58.26**	83.88	90.34	**3.001**	**91.68**	**96.80**	**97.89**	**0.715**	**92.31**	97.52	98.39	0.779
Ours det.	55.86	83.32	90.02	3.141	90.49	96.39	97.66	0.810	91.21	97.52	98.51	0.610
Ours mean	56.50	83.56	90.18	3.105	90.65	96.45	97.69	0.794	91.48	97.50	98.50	0.583
Ours cycle	57.16	**84.19**	**90.61**	3.051	91.24	96.67	97.81	0.764	91.90	**97.67**	**98.61**	**0.541**

(b) Training domains

	BlendedMVS (train)				VKITTI2 (train)			
Method	@1↑	@3↑	@5↑	EPE↓	@1↑	@3↑	@5↑	EPE↓
RoMaV2	**73.61**	**87.22**	**91.82**	**2.000**	**94.37**	98.16	98.93	**0.390**
Ours det.	69.44	86.16	91.27	2.214	94.00	98.16	98.92	0.423
Ours mean	69.82	86.35	91.41	2.179	94.11	98.22	98.96	0.411
Ours cycle	70.73	86.87	91.74	2.142	94.29	**98.27**	**98.97**	0.397

E.2 Input resolution

Table 13a evaluates all seven populations with input and scoring at 1024. The K=32 mean exceeds RoMaV2 in PCK@1 on all seven and in EPE on all except ScanNet++. Deterministic inference is already ahead in PCK@1 on ScanNet++, Hypersim, FlyingThings3D, and UnrealStereo4K, and has much lower EPE than RoMaV2 on UnrealStereo4K (1.72 vs. 3.23 px). Means over fewer draws are in Table 14.

Native-1024 scores use a stricter pixel tolerance than native-640 scores, so Table 13b transports the 1024 predictions to the 640 grid before scoring. On this grid, FlowMatcher's advantage at 1024 input persists on every population in PCK@1. However, RoMaV2 degrades on every population when moved from 640 to 1024 input, and FlowMatcher degrades less but improves only on MegaDepth; RoMaV2 at 640 remains the most accurate configuration in PCK@1 on all seven populations. The resolution result of Sec. 4.1 is thus a statement about robustness to input resolution, not about higher accuracy.

E.3 Test-time budgets and inference-time interventions

Integration steps N and draws K act differently: N changes each prediction, while K changes the mean and the covariance estimate. Table 14a gives the complete dense grids. At 640, both have small effects on accuracy, and K saturates by $K{\approx}8$. At 1024, where the model operates away from its training resolution, both matter more: on Hypersim, PCK@1 rises from 46.48 (N=4, K=1) to 49.55 (N=8, K=32). Pose is insensitive to K for the mean prediction (Table 14b).

Inference-time interventions. Three interventions probe what the trained refiner relies on (Table 15); none replaces retraining without the corresponding input. Fixing the time input to zero while keeping the integration schedule (panel a) has no effect at N=1, where $t = 0$ anyway, and increasingly hurts with more steps: at N=8, MegaDepth PCK@1 drops from 60.71 to 56.46, and the gap widens when the coarse warp is perturbed. The refiner therefore uses time to adapt its updates along the trajectory. Replacing the scale-conditioning channel by a constant with zero source noise in both arms (panel b) costs 0.40 PCK@1 (95% interval $[-0.70, -0.12]$), a small but resolved effect. Replacing the heteroscedastic source by a spatially constant scale at inference (panel c), with everything else fixed, lowers PCK@1 at the same K on six populations and ties on UnrealStereo4K, where the flat source has lower EPE. Because the model was trained with heteroscedastic noise and still receives the scale map as conditioning, this tests how the trained model uses its source; the training-time comparison is Table 6.

Table 13: **Input resolution.** RoMaV2 without its high-resolution pass; ours $N{=}8$. (a) Input and scoring at 1024: PCK@d (%) and EPE (px); bold: best of the three per population and metric. (b) The same 1024-input predictions scored on the 640 grid; 640-input scores are in Tables 1 and 12. Bold: better model.

(a) Native 1024

	RoMaV2				Ours, deterministic				Ours, mean $K{=}32$			
Population	@1↑	@3↑	@5↑	EPE↓	@1↑	@3↑	@5↑	EPE↓	@1↑	@3↑	@5↑	EPE↓
MegaDepth	63.16	88.43	93.73	1.893	62.51	90.26	94.96	1.834	**64.28**	**90.90**	**95.32**	**1.764**
ScanNet++ v2	26.06	59.95	75.75	**5.733**	27.56	**61.75**	**76.28**	5.784	**28.02**	61.00	75.48	5.891
Hypersim	42.84	68.32	78.31	7.128	47.38	**72.27**	**80.48**	6.895	**49.55**	72.16	80.24	**6.891**
Aerial cross	14.20	43.36	60.62	10.464	13.32	43.59	61.26	10.592	**14.53**	**44.92**	**61.96**	**10.419**
Aerial mixed	38.73	70.73	81.75	5.121	37.98	72.55	82.95	5.208	**39.67**	**73.56**	**83.47**	**5.092**
FlyingThings3D	77.74	91.57	95.26	1.577	82.32	93.92	96.27	1.465	**83.64**	**94.12**	**96.38**	**1.400**
UnrealStereo4K	85.85	92.87	94.92	3.227	86.93	**94.19**	**96.01**	**1.721**	**86.95**	93.93	95.80	1.901

(b) 1024 input, scored on the 640 grid

	PCK@1 (%)↑		EPE (px)↓	
Population	RoMaV2	Ours	RoMaV2	Ours
MegaDepth	77.31	**79.80**	1.174	**1.088**
ScanNet++ v2	39.37	**41.60**	**3.539**	3.629
Hypersim	54.20	**60.41**	4.321	**4.172**
Aerial cross	24.81	**25.80**	6.512	**6.470**
Aerial mixed	53.96	**56.34**	3.180	**3.151**
FlyingThings3D	84.86	**89.64**	0.973	**0.856**
UnrealStereo4K	89.25	**90.44**	2.012	**1.181**

E.4 HPatches and pose details

HPatches. Table 16a gives the individual corruptions behind the family averages of Table 2, on identical corrupted images for both models. FlowMatcher has lower AEPE on 10 of 13 corruptions and on average. Its clear weakness is pixel-level noise (Gaussian, impulse, shot). A plausible cause is the stride-one correlation window, which relies on the fine features that pixel noise corrupts most. The homography gains of Table 2 hold on both the illumination and viewpoint subsets (Table 16b).

Pose with fewer draws and cycle selection. Table 14b reports pose for every K; for the mean prediction, pose is insensitive to K. For the matched 640 comparison over the same 32 draws, cycle-minus-mean AUC@5 differences with 95% paired intervals are $0.72\,[0.12, 1.30]$ on MegaDepth, $0.38\,[-0.04, 0.75]$ on ScanNet, and $0.21\,[0.04, 0.30]$ on ETH3D.

Baselines on ETH3D. All local ETH3D rows in Table 3 share the same pairs, cameras, solver, and five seeds; correspondences are mapped to native image coordinates before normalization. Original RoMa uses outdoor weights at $560 \to 864$ and DKM its native $540 \times 720 \to 864 \times 1152$, each sampling 5,000 matches with its own balanced sampler per trial. LoFTR uses outdoor weights on grayscale images at 1024 long edge. SuperPoint extracts up to 4,096 keypoints at 1024 long edge, matched by LightGlue without adaptive pruning. MASt3R uses 512 long-edge full-frame images with reciprocal matching at stride eight (a full-frame adaptation of the authors' center-crop loader). Sparse methods keep at most 5,000 highest-confidence matches. MegaDepth and ScanNet values for these five methods are published numbers.

E.5 Uncertainty ranking and calibration

Table 17a extends Table 5 with scene-bootstrap intervals, intermediate K, PCK-based AUSE, and the retained EPE after removing half of the pixels. Spread improves monotonically with K on every population and metric. On ScanNet++ and Hypersim, spread with $K \geq 16$ beats the trained precision head on every metric, and EPE-AUSE intervals no longer overlap from $K{=}8$ on. On MegaDepth, $K{=}32$ spread and the trained precision head are statistically tied in EPE AUSE, and

Table 14: **Integration steps N and draws K.** (a) Dense: PCK@1 (%) / EPE (native px) of the K-sample mean, nested draws; 640 and 1024 have different pixel tolerances. (b) Pose AUC (%) of the K-sample mean, settings of Table 3; bottom: mean vs. cycle over the same 32 draws at 640. Bold: best per column and block.

(a) Dense: PCK@1 (%)↑ / EPE (px)↓

		MegaDepth		ScanNet++	
N	K	640	1024	640	1024
4	1	78.11 / 1.181	60.55 / 1.868	46.35 / 3.251	25.89 / 6.008
4	2	78.31 / 1.173	62.13 / 1.822	46.54 / 3.243	26.63 / 5.952
4	4	78.41 / 1.169	62.89 / 1.800	46.62 / 3.240	26.98 / 5.925
4	8	78.47 / 1.167	63.30 / 1.787	46.69 / 3.238	27.21 / 5.911
4	16	78.50 / 1.165	63.52 / 1.780	46.74 / 3.236	27.31 / 5.901
4	32	78.51 / 1.165	63.63 / 1.778	46.76 / 3.235	27.35 / 5.897
8	1	78.44 / 1.170	62.77 / 1.817	46.72 / 3.239	27.18 / 5.972
8	2	78.56 / 1.165	63.56 / 1.788	46.83 / 3.233	27.64 / 5.930
8	4	78.63 / 1.161	63.94 / 1.776	46.88 / 3.231	27.83 / 5.910
8	8	78.67 / 1.160	64.13 / 1.768	46.92 / 3.229	27.97 / 5.901
8	16	78.69 / **1.159**	64.24 / **1.764**	46.97 / 3.228	28.01 / 5.894
8	32	**78.70** / **1.159**	**64.28** / **1.764**	**46.99** / **3.228**	**28.02** / **5.891**

		Hypersim		Aerial cross	
N	K	640	1024	640	1024
4	1	63.08 / 4.218	46.48 / 7.061	28.68 / 6.365	13.11 / 10.603
4	2	63.27 / 4.206	47.63 / 6.990	28.90 / 6.354	13.65 / 10.518
4	4	63.37 / 4.198	48.21 / 6.948	28.95 / 6.345	13.91 / 10.462
4	8	63.40 / 4.196	48.51 / 6.922	28.98 / 6.339	14.04 / 10.439
4	16	63.43 / 4.194	48.69 / 6.910	29.00 / 6.335	14.14 / 10.425
4	32	63.44 / 4.194	48.75 / 6.905	29.03 / 6.334	14.16 / **10.418**
8	1	63.62 / 4.201	48.43 / 7.001	29.05 / 6.359	13.92 / 10.575
8	2	63.75 / 4.192	48.99 / 6.954	29.16 / 6.352	14.19 / 10.507
8	4	63.81 / 4.184	49.28 / 6.923	29.22 / 6.343	14.37 / 10.459
8	8	63.83 / 4.182	49.44 / 6.905	29.25 / 6.338	14.47 / 10.436
8	16	**63.84** / **4.180**	49.52 / 6.895	29.27 / 6.334	14.51 / 10.424
8	32	**63.84** / **4.180**	**49.55** / **6.891**	**29.29** / **6.331**	**14.53** / 10.419

		Aerial mixed		FlyingThings3D		UnrealStereo4K	
N	K	640	1024	640	1024	640	1024
4	1	55.74 / 3.139	36.72 / 5.251	90.56 / 0.796	82.99 / 1.449	91.38 / 0.587	86.20 / 2.171
4	2	55.99 / 3.124	38.00 / 5.181	90.60 / 0.792	83.47 / 1.417	91.41 / 0.583	86.48 / 2.115
4	4	56.11 / 3.117	38.69 / 5.143	90.62 / 0.790	83.70 / 1.400	91.41 / 0.581	86.61 / 2.088
4	8	56.18 / 3.113	39.02 / 5.122	90.64 / 0.789	83.81 / 1.391	91.41 / **0.580**	86.68 / 2.069
4	16	56.20 / 3.111	39.19 / 5.111	**90.65** / **0.788**	83.87 / 1.386	91.41 / **0.580**	86.71 / 2.060
4	32	56.23 / 3.110	39.29 / 5.105	**90.65** / **0.788**	**83.89** / **1.384**	91.42 / **0.580**	86.73 / 2.054
8	1	56.17 / 3.129	38.33 / 5.204	90.60 / 0.799	83.29 / 1.427	91.45 / 0.588	86.81 / 1.967
8	2	56.33 / 3.117	39.01 / 5.152	90.62 / 0.796	83.48 / 1.413	**91.48** / 0.586	86.89 / 1.932
8	4	56.41 / 3.111	39.36 / 5.122	90.64 / 0.795	83.57 / 1.405	**91.48** / 0.584	86.92 / 1.918
8	8	56.45 / 3.108	39.54 / 5.105	**90.65** / 0.794	83.61 / 1.403	**91.48** / 0.584	86.94 / 1.907
8	16	56.48 / 3.106	39.62 / 5.097	**90.65** / 0.794	83.63 / 1.400	**91.48** / 0.584	**86.95** / 1.903
8	32	**56.50** / **3.105**	**39.67** / **5.092**	**90.65** / 0.794	83.64 / 1.400	**91.48** / 0.583	**86.95** / **1.901**

(b) Relative pose

	MegaDepth			ScanNet			ETH3D		
Prediction	@5° ↑	@10° ↑	@20° ↑	@5° ↑	@10° ↑	@20° ↑	@5° ↑	@10° ↑	@15° ↑
Deterministic	60.01	74.86	85.27	33.20	55.66	73.12	83.51	89.16	91.46
Mean K=4	60.12	75.01	85.40	**34.12**	**56.61**	**73.81**	**83.91**	**89.46**	**91.70**
Mean K=8	**60.50**	**75.14**	**85.47**	33.05	55.69	73.14	83.87	89.41	91.63
Mean K=16	60.37	75.05	85.41	33.56	55.88	73.35	83.84	89.39	91.65
Mean K=32	60.36	75.03	85.29	33.17	55.81	73.31	83.86	89.41	91.65
Mean K=32, 640	58.83	73.87	84.53	33.50	56.12	73.57	83.79	89.73	92.14
Cycle K=32, 640	**59.55**	**74.41**	**84.85**	**33.89**	**56.47**	**73.80**	**84.00**	**89.85**	**92.20**

Table 15: **Inference-time interventions** on the trained model; input and scoring 1024. (a) Time input, 200 pairs per population: t=0 feeds zero time at every step without retraining; corruption adds an iid perturbation with the stated RMS to the coarse warp. (b) Scale channel, MegaDepth, 500 pairs, N=8, zero source noise in both arms. (c) Source, N=8, mean of K draws: flat replaces the heteroscedastic source by a spatially constant scale. Bold: better arm.

(a) Time input

	MegaDepth				ScanNet++ v2			
	PCK@1 (%)↑		EPE (px)↓		PCK@1 (%)↑		EPE (px)↓	
Setting	time	t=0	time	t=0	time	t=0	time	t=0
Clean, N=1	53.09	53.09	2.165	2.165	22.11	22.11	5.783	5.783
Clean, N=2	**58.07**	56.36	**1.978**	2.009	**26.95**	24.83	**5.470**	5.591
Clean, N=4	**60.05**	56.41	**1.907**	1.973	**27.77**	24.94	**5.443**	5.676
Clean, N=8	**60.71**	56.46	**1.882**	1.957	**27.83**	24.55	**5.487**	5.830
Corruption 0 px, N=8	**60.80**	58.41	**1.860**	1.920	**27.23**	24.90	**5.742**	6.158
Corruption 4 px, N=8	**59.56**	55.03	**1.928**	2.026	**25.66**	22.89	**5.938**	6.340
Corruption 8 px, N=8	**54.25**	43.00	**2.316**	2.578	**20.63**	16.44	**6.737**	7.117
Corruption 16 px, N=8	**32.15**	19.27	**6.261**	6.598	**10.22**	6.97	**10.919**	11.002

(b) Scale channel

Scale channel	PCK@1 (%)↑	PCK@3 (%)↑	PCK@5 (%)↑	EPE (px)↓
Predicted σ	**62.51**	90.26	94.96	**1.834**
Constant	62.11	**90.28**	**95.03**	1.838

(c) Source scale

Source	MegaDepth	ScanNet++	Hypersim	Aer. cross	Aer. mixed	FT3D	US4K
PCK@1 (%)↑							
Heteroscedastic, K=4	**63.94**	**27.83**	**49.28**	**14.37**	**39.36**	**83.57**	86.92
Flat, K=4	63.67	26.39	47.57	13.52	38.59	82.98	**86.93**
Heteroscedastic, K=8	**64.13**	**27.97**	**49.44**	**14.47**	**39.54**	**83.61**	86.94
Flat, K=8	63.76	26.51	47.64	13.58	38.67	82.99	86.94
EPE (px)↓							
Heteroscedastic, K=4	**1.776**	**5.910**	**6.923**	**10.459**	**5.122**	**1.405**	1.918
Flat, K=4	1.789	6.045	7.019	10.611	5.201	1.442	**1.694**
Heteroscedastic, K=8	**1.768**	**5.901**	**6.905**	**10.436**	**5.105**	**1.403**	1.907
Flat, K=8	1.783	6.034	7.004	10.593	5.187	1.440	**1.692**

Table 16: **HPatches details.** (a) AEPE (px) for individual severity-5 corruptions, grouped into the families of Table 2; input 640, forward, deterministic N=4; mean over the 13 corruptions. (b) Homography AUC (%) by subset; 1024, bidirectional, deterministic N=8. Bold: better model.

(a) Corruptions

Family	Corruption	RoMaV2	Ours
Weather	Snow	5.37	**4.50**
	Frost	27.19	**26.23**
	Fog	9.59	**7.76**
Digital	Brightness	7.25	**5.64**
	Contrast	7.56	**5.65**
	Pixelate	9.83	**7.10**
	JPEG	10.96	**7.96**
Blur	Defocus	9.61	**7.70**
	Motion	9.83	**8.05**
	Zoom	10.24	**8.99**
Pixel noise	Gaussian	**8.52**	8.84
	Impulse	**8.20**	8.54
	Shot	**8.05**	8.50
Mean (13 corruptions)		10.17	**8.88**

(b) Subsets

Subset	AUC↑	RoMaV2	Ours
Illumination	@3	80.70	**81.19**
	@5	87.96	**88.30**
	@10	93.79	**93.97**
Viewpoint	@3	63.39	**64.06**
	@5	74.28	**74.94**
	@10	84.71	**85.05**

Table 17: **Uncertainty details.** (a) Ranking our deterministic prediction at 1024, $N{=}8$; all metrics ↓. EPE AUSE with scene-bootstrap 95% intervals; "–": point estimate in Table 5. Trained head: $\mathrm{tr}(P^{-1})$ from the precision head of Sec. 4.3. (b) Coverage (%) of nominal 90% regions (ideal: 90) and calibration MAE (pp) over 50–99% levels. Left: 640, $N{=}8$, $K{=}32$. Right: ranges over 16 separately fitted settings per population (two resolutions, two step counts, four draw counts), 112 in total.

(a) Ranking

Signal	EPE AUSE	95% CI	PCK@1 AUSE	PCK@5 AUSE	EPE@50% (px)
MegaDepth					
Overlap	–	[0.2389, 0.2639]	0.1594	0.0196	1.344
Trained head	–	[0.1989, 0.2061]	**0.1427**	0.0125	**1.165**
Spread $K{=}4$	–	[0.2364, 0.2489]	0.1665	0.0164	1.286
Spread $K{=}8$	0.2188	[0.2161, 0.2215]	0.1535	0.0139	1.221
Spread $K{=}16$	0.2072	[0.2056, 0.2089]	0.1472	0.0128	1.184
Spread $K{=}32$	–	[0.2011, 0.2021]	0.1435	**0.0124**	1.174
Overlap + spread $K{=}4$	0.2237	[0.2209, 0.2265]	0.1572	0.0139	1.223
ScanNet++					
Overlap	–	[0.4111, 0.4731]	0.2516	0.1560	4.693
Trained head	–	[0.2482, 0.2781]	0.1397	0.0732	4.043
Spread $K{=}4$	–	[0.2365, 0.2565]	0.1540	0.0840	3.730
Spread $K{=}8$	0.2209	[0.2107, 0.2314]	0.1417	0.0755	3.559
Spread $K{=}16$	0.2092	[0.1991, 0.2196]	0.1361	0.0722	3.484
Spread $K{=}32$	–	[0.1937, 0.2142]	**0.1331**	**0.0707**	**3.459**
Overlap + spread $K{=}4$	0.2331	[0.2225, 0.2434]	0.1538	0.0816	3.559
Hypersim					
Overlap	–	[0.2732, 0.3193]	0.1702	0.0744	4.146
Trained head	–	[0.2194, 0.2463]	0.1194	0.0472	4.039
Spread $K{=}4$	–	[0.2003, 0.2301]	0.1213	0.0512	4.031
Spread $K{=}8$	0.1876	[0.1738, 0.2014]	0.1040	0.0444	3.834
Spread $K{=}16$	0.1761	[0.1626, 0.1898]	0.0961	0.0417	3.792
Spread $K{=}32$	–	[0.1570, 0.1842]	**0.0924**	**0.0405**	3.768
Overlap + spread $K{=}4$	0.1932	[0.1796, 0.2067]	0.1133	0.0454	**3.724**

(b) Calibration

	640, $N{=}8$, $K{=}32$				Range over settings		
	Coverage@90		MAE (pp)		Coverage@90		MAE (pp)
Population	Raw	Conformal	scalar↓	t↓	Raw	Conformal	t↓
MegaDepth	7.6	90.3	9.58	**0.14**	7.57–56.49	90.25–90.41	0.07–0.71
ScanNet++	4.7	90.4	10.98	**0.29**	4.71–39.70	90.26–90.38	0.19–1.89
Hypersim	11.1	90.4	12.25	**0.36**	11.09–57.35	90.28–90.45	0.12–1.34
Aerial cross	4.9	90.3	10.87	**0.20**	4.89–36.46	90.31–90.43	0.14–1.49
Aerial mixed	6.3	90.3	9.29	**0.50**	6.31–48.65	90.26–90.48	0.17–0.71
FT3D	9.7	90.4	8.93	**0.57**	9.69–72.28	90.34–90.52	0.21–0.59
US4K	10.3	90.5	11.10	**0.24**	10.28–80.49	90.42–90.52	0.21–1.59

the head is better in PCK@1 AUSE and retained EPE; note that MegaDepth has only two test scenes. Combining matchability with $K = 4$ spread improves EPE-AUSE, although $K = 8$ spread alone remains better on this metric.

Figure 5 and Table 17b extend the calibration analysis to 640 and to all seven populations. Three findings hold throughout. First, raw coverage is far below nominal (5–11% at 640 with $K{=}32$, and at most about 80% in any of the 112 settings). Second, the conformal scalar achieves 90.25–90.52% held-out coverage in every setting, but is miscalibrated at other levels (MAE 9–12 pp). Third, the Student-t model brings MAE below 2 pp in every setting, showing that the *shape* of the sample covariance, not only its ranking, carries information about the error distribution once heavier tails are allowed.

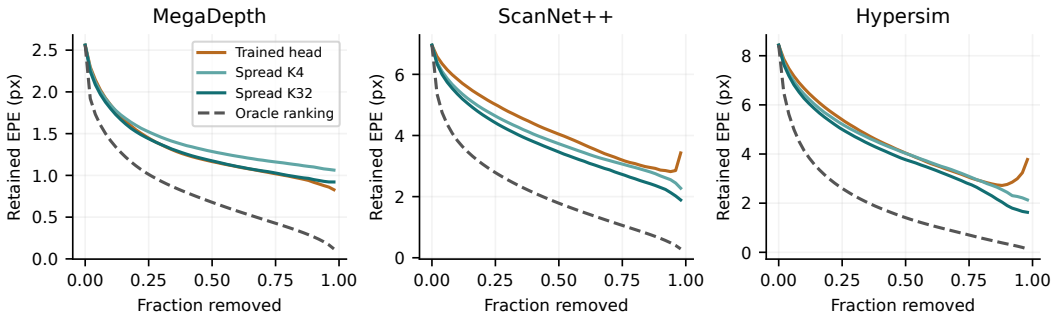


Figure 4: **Sample spread identifies inaccurate correspondences.** Retained EPE as the most uncertain pixels are removed, for the deterministic predictions of Table 5 (1024, N=8, 500 pairs per population). Dashed: oracle ranking by true error.

The limits are equally clear. Calibration is marginal: at 1024, pixels with the highest spread on the synthetic populations can under-cover by up to 19 pp, and on MegaDepth 18% of images have below 80% coverage despite 90% marginal coverage. Calibration also depends on setting: a scale fitted on MegaDepth at 1024 gives 70.5–95.1% coverage on the other populations, and a scale fitted at 1024 gives 58.9–84.1% at 640. Fitting at K=32 transfers well across K (90.2–92.2%). Coverage calibration is not required for the weighting gains of Table 4: raw and calibrated covariances differ by a global scale and a small isotropic floor, and a global scale leaves an unregularized weighted least-squares optimum unchanged. Calibration matters for the inlier threshold (Sec. 4.2).

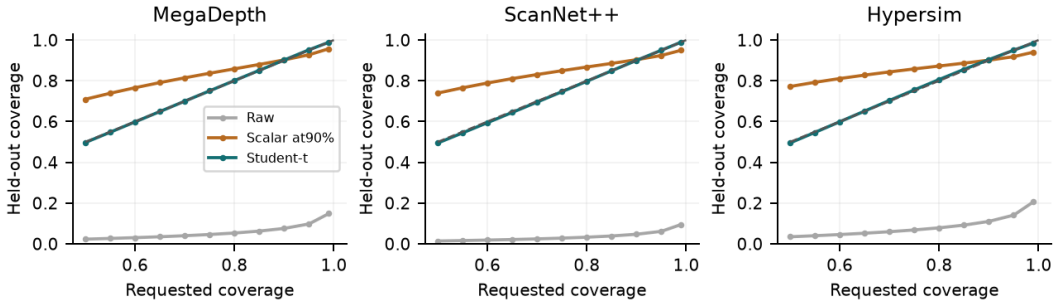


Figure 5: **Held-out reliability at 640, N=8, K=32.** Raw regions, the conformal scalar fitted at 90%, and the Student-t model; points are mean empirical coverage over held-out splits.

E.6 Runtime and memory

Table 18: **Runtime and memory** of a single prediction on an A100, batch one, 50 calls after 10 warm-ups; ours FP32, RoMaV2 FP32; forward direction except the bidirectional HR row; decoding excluded. Latency in ms (multiple of RoMaV2 without HR at the same input) and peak allocated memory in GiB.

	Latency (ms)↓			Memory (GiB)↓		
Model	640	800	1024	640	800	1024
RoMaV2, no HR	**94.2** (1.00×)	**130.5** (1.00×)	**220.2** (1.00×)	3.20	4.44	6.66
Ours, N=1	111.9 (1.19×)	158.6 (1.22×)	271.0 (1.23×)	**1.81**	**2.12**	**2.71**
Ours, N=4/4/2 per stage*	215.0 (2.28×)	319.3 (2.45×)	544.7 (2.47×)	1.82	2.14	2.74
Ours, N=4	245.6 (2.61×)	366.9 (2.81×)	622.0 (2.82×)	1.82	2.14	2.74
Ours, N=8	427.6 (4.54×)	644.4 (4.94×)	1096.0 (4.98×)	1.82	2.14	2.74
RoMaV2 + HR, bidirectional	286.4	341.0	462.4	6.56	6.65	6.87

* Steps at strides 4/2/1.

Table 18 times a single prediction at three input resolutions. One integration step per stage adds 19–23% to RoMaV2's latency, and latency grows linearly with N. FlowMatcher's refiners allocate less memory than RoMaV2's at every resolution, and memory does not grow with N. Draws run sequentially, so latency grows about linearly with K while peak memory stays constant (Table 7). The table does not time cycle selection, covariance construction, or pose estimation. Since the cost of sampling is dominated by the $3NK$ refiner calls, the natural routes to lower cost are fewer steps for later draws, sampling only at pixels where the first draws disagree, and distillation; we leave these to future work.

F QUALITATIVE RESULTS

The following plates show predictions of the evaluated model. Full-field examples use 1024×1024 inputs and query-level examples 640×640, both with $K{=}32$ and $N{=}8$. Examples are illustrative and were selected to show specific behaviors; they are not representative of average performance.

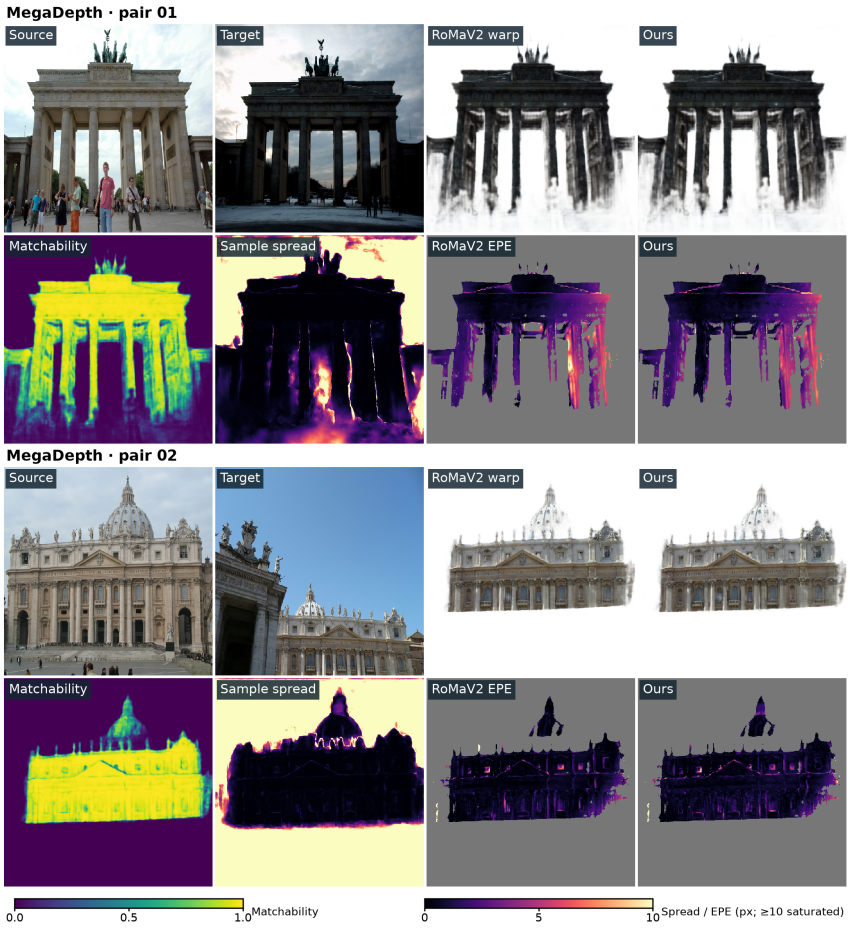


Figure 6: **Outdoor warps, matchability, and uncertainty** on two MegaDepth pairs. Top rows: source, target, and the target sampled at each method's predicted correspondences (ours at the sample mean), multiplied by our predicted covisibility; no ground-truth masking is applied. Bottom rows: matchability, sample spread $\sqrt{\mathrm{tr}\, S}$, and each method's endpoint error. Spread and error share a 0–10 px scale; gray marks missing ground truth. Spread is high near occlusion boundaries and outside the covisible region.

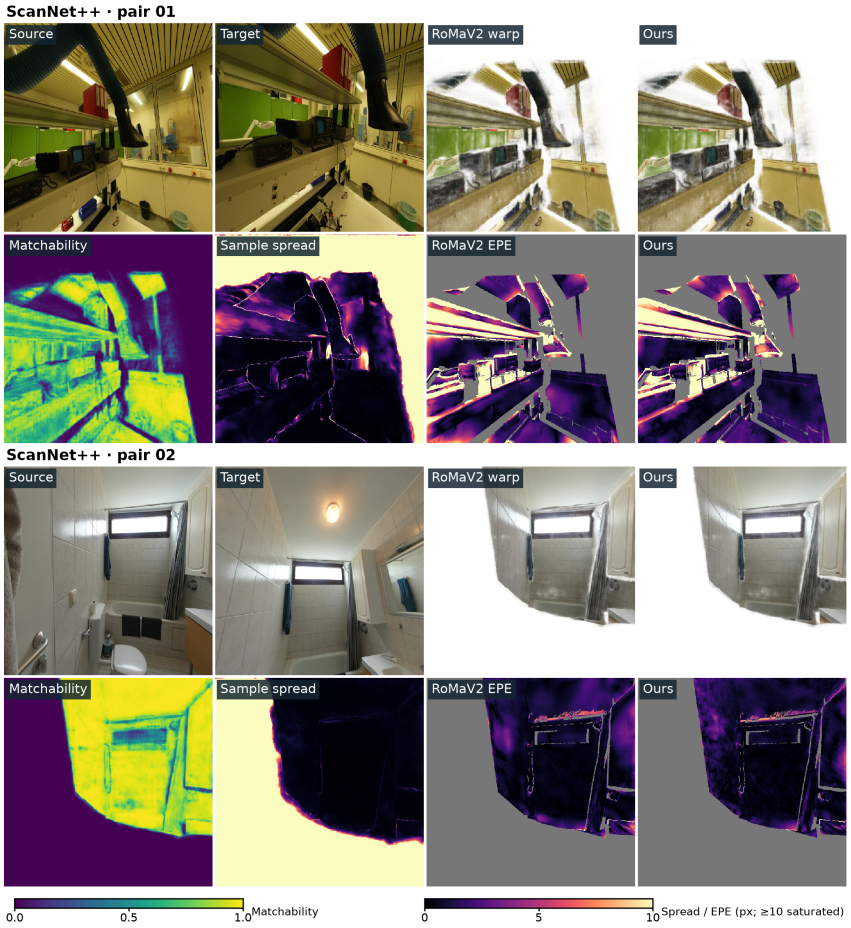


Figure 7: **Indoor warps, matchability, and uncertainty** on two ScanNet++ pairs, with the conventions of Fig. 6. Matchability and spread capture different properties: a pixel can be confidently matchable and still have uncertain localization, and a concentrated set of samples can still be wrong.

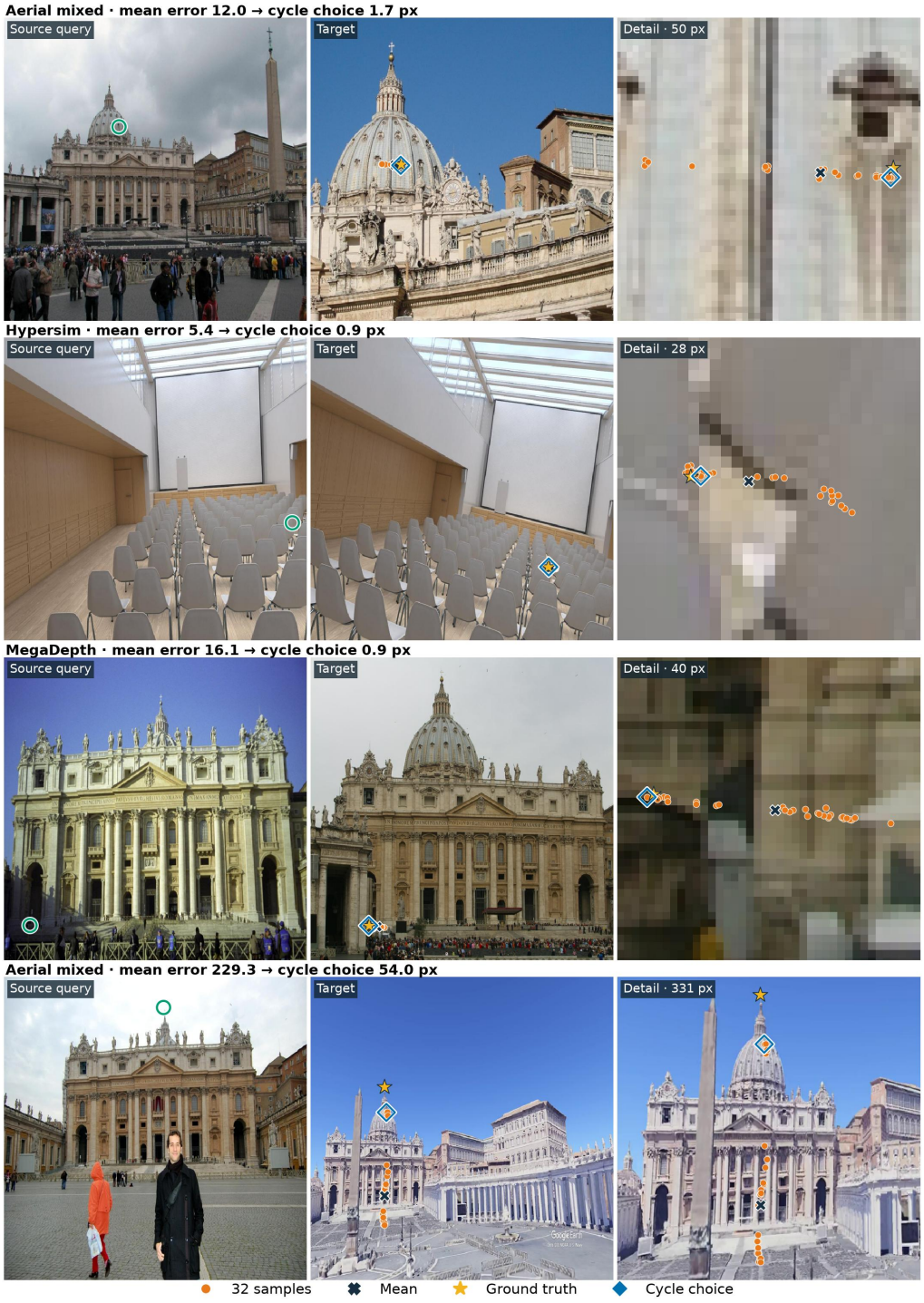


Figure 8: **Cycle selection recovering from a misleading mean.** Each row shows the source query, the target, and a target crop with all 32 endpoints, their mean, ground truth, and the candidate chosen by Eq. 8. Errors are in 640-grid pixels. The samples form separated groups and the mean falls between them; in the first three rows cycle selection picks a candidate near ground truth. Even a large improvement need not give an accurate match (last row). Crops have independent widths.

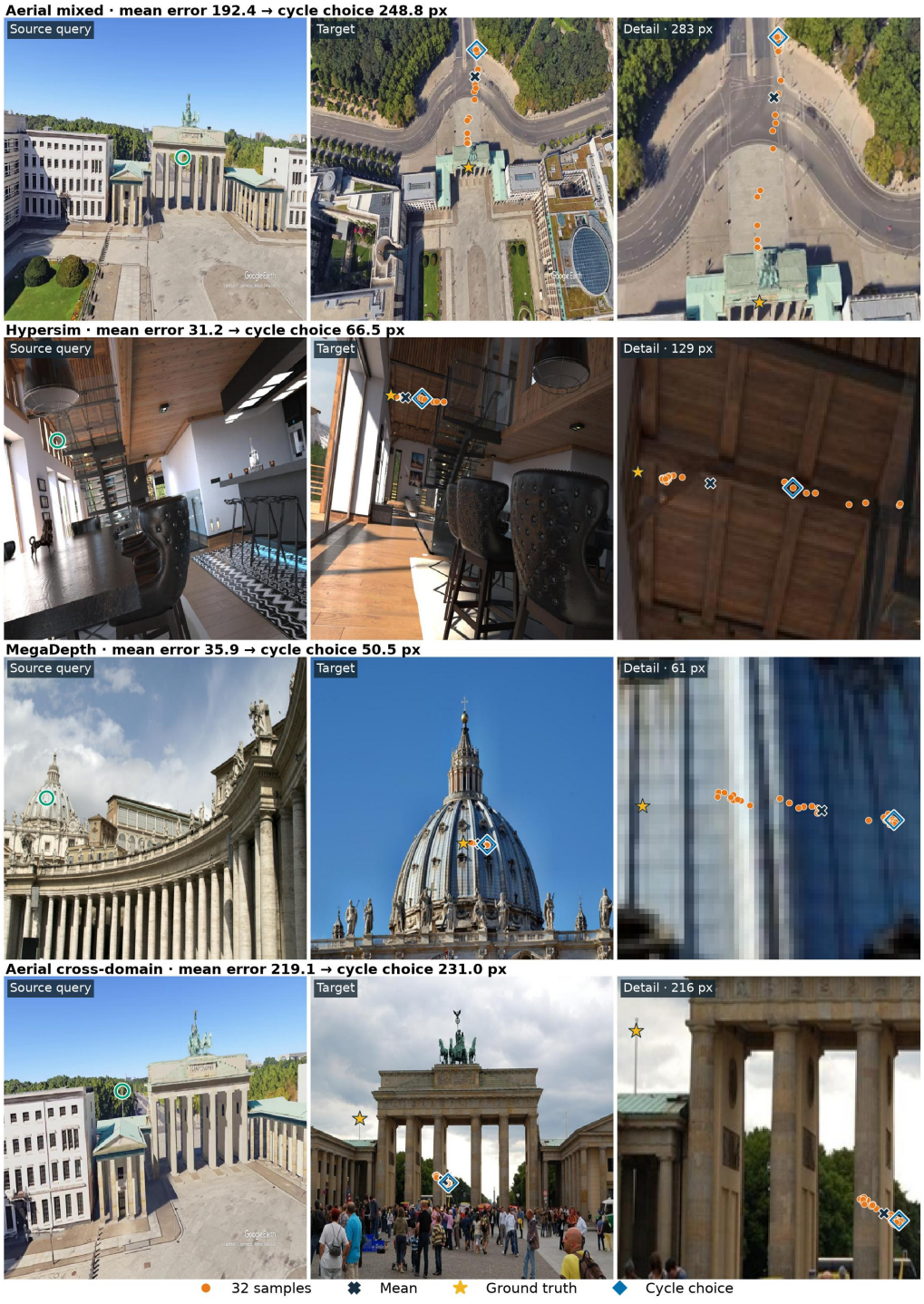


Figure 9: **Failure cases of cycle selection**, with the visualization of Fig. 8. Here the selected candidate is worse than the mean: several groups contain inaccurate matches, and reverse consistency favors the wrong one. Ground truth is shown for illustration and is not used by the selector.