# BEYOND INLIER CONFIDENCE: LEARNING RESIDUAL QUANTILES FOR CORRESPONDENCE PRUNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Learned two-view correspondence pruning commonly uses scalar inlier scores to guide geometric estimation. However, correspondences with similar probabilities of satisfying an inlier threshold can have different residual distributions. We propose Residual Quantile Regression (RQR), which augments inlier classification with conditional geometric residual quantiles. Using a pinball objective, RQR learns ordered quantiles of the log epipolar residual for retained correspondences, providing a compact description of residual magnitude and spread. These quantiles guide correspondence learning during training; in the inference-time realization, RQR-F, they condition features through gated updates, while their spreads inform weights for differentiable weighted eight-point estimation. This formulation makes residual statistics explicit in correspondence representations and geometric weighting. Experiments on relative pose and homography estimation demonstrate the effectiveness of residual quantile learning across pruning architectures and geometric tasks.

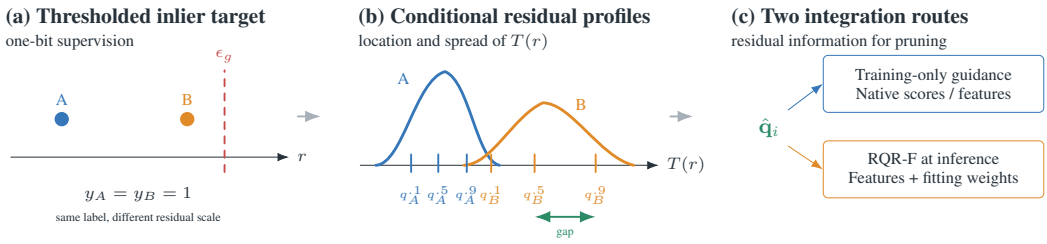



Figure 1: **From inlier labels to residual quantiles.** Matches sharing an inlier label can have different residual profiles.

## 1 INTRODUCTION

Two-view geometry requires deciding which tentative correspondences should influence an estimate. Learned pruners model agreement among matches and predict scores for geometric fitting (Yi et al., 2018; Zhang et al., 2019; Zhao et al., 2021). Their classification targets usually threshold a geometric residual. Such labels omit residual magnitude within each class, although this information can inform fitting together with the spatial configuration and relative weights of the correspondences.

Even an exact inlier probability specifies the residual distribution at only one threshold. Distributions with the same probability below that threshold can differ in their medians and upper quantiles, as shown in Figure 1. Binary supervision leaves these quantities unspecified. RQR supervises them explicitly, complementing the host's classification and geometric objectives. This distinction matters when confidence scores are converted into fitting weights: similar threshold confidence need not imply similar residual spread. Learning several residual levels gives the pruner an explicit description of this difference while preserving the role of contextual evidence in identifying geometric consensus.

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.

We introduce Residual Quantile Regression (RQR) to predict ordered levels of a transformed geometric residual. A single regression target describes location; equal medians can still accompany different upper tails. Multiple quantile targets constrain both location and tail extent without specifying a parametric density. Auxiliary heads guide native scores or supervise shared features during training and are discarded at inference. RQR-F retains the descriptor to refine features and geometric weights.

The experiments ask where explicit residual information helps: in training existing pruners, in fitting geometry at inference, and under changes of scene, feature, or task. DeMatch++ improves weighted pose AUC at all three tolerances in all six evaluated dataset–feature settings. Continuous-target controls establish the value of residual supervision while revealing its dependence on placement. A separate fixed-host study evaluates whether retaining the quantile descriptor is useful during inference.

**Contributions.** We formulate an ordered residual descriptor that represents location and tail extent beyond binary inlier targets. We develop training-time score guidance and shared-feature supervision, and an inference-time realization, RQR-F, that conditions features and fitting weights. Paired evaluations across six pruners show gains at all weighted pose AUC tolerances for five hosts. Continuous-target controls, solver diagnostics, and frozen transfer characterize the roles of supervision placement, estimation, and input features.

## 2 Related Work

**From local features to geometric consensus.** Correspondence pipelines connect feature extraction and matching to geometric estimation. SIFT and SuperPoint provide local features (Lowe, 2004; DeTone et al., 2018); SuperGlue and LoFTR learn interactions to establish matches (Sarlin et al., 2020; Sun et al., 2021). LightGlue adapts depth through early exit and width through point pruning (Lindenberger et al., 2023). Its confidence controls matching computation; our residual targets supervise downstream geometric consistency. Recent matchers incorporate multiple views, cascaded priors, or geometric and semantic features (Zhang et al., 2025a; Chen et al., 2025; Xue et al., 2025). Local motion consistency (Bian et al., 2017), RANSAC, and local optimization filter or fit candidate matches (Fischler & Bolles, 1981; Chum et al., 2003). Learned estimation also predicts correspondence weights and hypothesis sampling probabilities (Ranftl & Koltun, 2018; Brachmann & Rother, 2019). RQR operates on given matches rather than replacing their construction.

**Learning correspondence context.** Correspondence encoders model unordered sets through aggregation and interactions (Zaheer et al., 2017; Lee et al., 2019; Zhang et al., 2022). Context normalization and ordered pooling provide early mechanisms for exchanging correspondence evidence (Yi et al., 2018; Zhang et al., 2019); attentive normalization and channel context refine these representations (Sun et al., 2020; Zhong et al., 2021). Graph methods capture consensus during pruning (Zhao et al., 2021; Dai et al., 2022; Lin et al., 2025), while bilateral context and motion modeling offer complementary designs (Miao et al., 2024; Zhang & Ma, 2024; Lu et al., 2025; Zhang et al., 2025b; Le & Ma, 2026). Attention combines local and global evidence (Liu et al., 2025), and LeCoT revisits the pruning architecture (Dai et al., 2026). These methods determine how matches exchange information. RQR adds an explicit objective for the geometric information their features should represent.

**Confidence and residual distributions.** Geometric reliability has been modeled beyond binary inlier confidence. PDC-Net+ predicts dense correspondence uncertainty with a probabilistic flow model (Truong et al., 2023). Noise-aware consensus learning predicts outlier status and inlier displacement noise for weighted estimation (Moran et al., 2024). RQR instead predicts scalar residual quantiles under the reference epipolar geometry, using pinball supervision without choosing a parametric density (Koenker & Bassett, 1978). The distinction lies in the predicted geometric quantity and its coupling to correspondence learning. Coordinate correction modifies the observations supplied to a solver; residual prediction describes their geometric consistency while retaining the observed coordinates. Moreover, a geometric loss evaluates a fitted matrix jointly across matches, whereas RQR supplies targets at individual retained correspondences. This additional supervision makes several residual levels explicit without replacing the host's consensus mechanism or requiring a new fitting algorithm.

# 3 METHOD

A match-set encoder supplies context-aware features; RQR specifies the geometric quantity learned from them, as illustrated in Figure 2. We first define residual quantiles and their supervision, then explain how they guide existing pruners during training. RQR-F extends the same representation to inference by conditioning features and geometric weights.

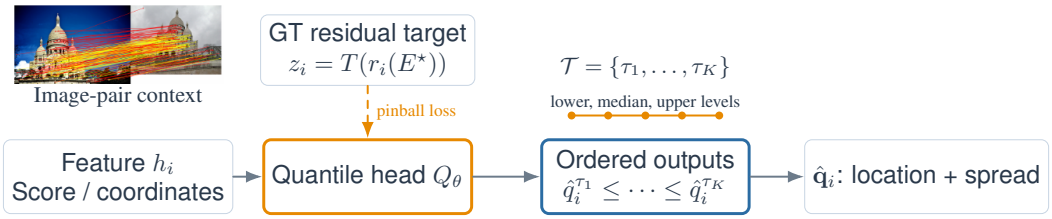



Figure 2: **Learning residual quantiles.** Ordered residual levels link host features to geometric supervision.

## 3.1 RESIDUAL TARGET AND CONDITIONAL QUANTILES

Let $\mathcal{C} = \{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^N$ contain correspondences in normalized homogeneous coordinates. The host produces feature $h_i$ and scalar score $s_i$. Given ground-truth essential matrix $E^\star$, we use the symmetric epipolar residual

$$r_i = (\mathbf{x}_i'^\top E^\star \mathbf{x}_i)^2 \left[ \frac{1}{\|(E^\star \mathbf{x}_i)_{1:2}\|_2^2 + \delta} + \frac{1}{\|(E^{\star\top} \mathbf{x}'_i)_{1:2}\|_2^2 + \delta} \right]. \tag{1}$$

The stabilizer $\delta$ is set to $10^{-15}$. The native inlier label is $y_i = \mathbf{1}\{r_i < \epsilon_g\}$, with $\epsilon_g = 10^{-4}$. RQR preserves this label and adds a transformed continuous target $z_i = T(r_i)$. Log compression reduces the residual dynamic range; the specific transform follows the coupling below.

For ordered levels $\mathcal{T} = \{\tau_1 < \cdots < \tau_K\}$, a shared predictor maps the host representation to

$$Q_\theta(h_i, s_i, \mathbf{c}_i) = \hat{\mathbf{q}}_i, \qquad \hat{\mathbf{q}}_i = (\hat{q}_i^{\tau_1}, \ldots, \hat{q}_i^{\tau_K}), \tag{2}$$

where $\mathbf{c}_i$ contains the four nonhomogeneous coordinates; scores and coordinates are optional inputs. Each output estimates a quantile of $z = T(r)$ conditional on these inputs. The distribution is defined over training correspondences, each providing one residual measured against its reference geometry.

## 3.2 ORDERING AND PINBALL SUPERVISION

Training-side heads enforce ordering with positive increments: $\hat{q}_i^{\tau_k} = \hat{q}_i^{\tau_{k-1}} + \mathrm{softplus}(\ell_{ik}) + \varepsilon_{\mathrm{ord}}$ for $k > 1$. Score guidance uses a free base $\ell_{i1}$ and $\varepsilon_{\mathrm{ord}} = 10^{-4}$; shared-feature heads use a softplus base and $\varepsilon_{\mathrm{ord}} = 0$. RQR-F instead sorts three raw outputs. All use pinball supervision on the retained set $\mathcal{I}$:

$$\mathcal{L}_q = \frac{1}{|\mathcal{I}|K} \sum_{i \in \mathcal{I}} \sum_{\tau \in \mathcal{T}} \rho_\tau(z_i - \hat{q}_i^\tau), \tag{3}$$

where $\rho_\tau(u) = \max\{\tau u, (\tau - 1)u\}$ is the pinball loss. For fixed inputs, expected pinball loss is minimized by a conditional quantile (Koenker & Bassett, 1978). At a continuous conditional CDF, its derivative is $F_{z|X}(q) - \tau$, so different levels constrain different parts of the residual distribution. Finite capacity and joint geometric or ranking objectives can shift the learned outputs from this optimum. Ordering prevents crossings; empirical coverage is a separate property.

For strictly increasing continuous $T$, quantiles satisfy $Q_{T(r)|X}(\tau) = T(Q_{r|X}(\tau))$, where $X$ denotes predictor inputs. The score-guidance target additionally floors residuals at $10^{-8}$, making smaller residuals indistinguishable. Quantile gaps are interpreted on the transformed scale.

### 3.3 COUPLING RESIDUAL QUANTILES TO EXISTING PRUNERS

A classifier targets the probability of satisfying one residual threshold. RQR additionally supervises residual levels through either score guidance or shared features, while preserving the host's inference interface.

**Decision-local score guidance.** A width-16 head combines detached features, standardized scores, and coordinates with pooled mean context. It predicts five levels $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ of $z_i = \log \max(r_i, 10^{-8})$. The median and upper level define a residual-risk teacher

$$R_i = \exp(\hat{q}_i^{0.5}/2) + \beta\big[\exp(\hat{q}_i^{0.9}/2) - \exp(\hat{q}_i^{0.5}/2)\big]. \tag{4}$$

We set $\beta = 0.5$, averaging the median and upper square-root residual levels, and clip predicted log values to $[-24, 8]$ before exponentiation. OANet aligns standardized native scores with detached negative risk near the selection boundary using Smooth L1 loss. PointCN, DeMatch++, and GeoMoE use ranking guidance and incorporate the eight-point design-row norm into the risk. The teacher also receives a boundary-ranking loss; DeMatch++ and GeoMoE add geometric-risk ranking. Predictor inputs are detached, while separate guidance transfers the learned risk ordering to native scores. Appendix A specifies the losses and gradient paths. Inference retains the native score and solver.

**Shared-feature supervision.** For CLNet and BCLNet, auxiliary heads backpropagate into prediction-layer features. Three ordered outputs at $\{0.1, 0.5, 0.9\}$ predict $z_i = \log(1 + r_i/\epsilon_g)$, with targets gathered at the retained indices. CLNet averages the pinball losses from three interfaces, each averaged over its own matches and quantiles. Paired BCLNet supervises only its final interface. Both retain native losses and add $\lambda_q \mathcal{L}_q$ with $\lambda_q = 0.25$. The heads are removed at inference.

For progressive pruners, residual targets follow the retained feature indices. Supervision placement determines which representations learn residual structure. Both couplings discard their auxiliary heads at inference; RQR-F instead uses the predicted descriptor to condition features and weights, as described next.

### 3.4 RESIDUAL-CONDITIONED INFERENCE: RQR-F

RQR-F is an inference-time realization of residual quantile learning. A Transformer-based correspondence encoder provides retained features; the predicted residual descriptor then informs both feature refinement and geometric weighting. The component study evaluates these additions on a fixed host.

**Residual descriptor and feature conditioning.** The host combines local attention, order-aware pooling/unpooling, and two half-set selections. A shared $128 \to 3$ projection predicts sorted outputs at $\{0.1, 0.5, 0.9\}$ for $z_i = \log(r_i + \epsilon_0)$, where $\epsilon_0 = 10^{-6}$. Sorting is the full model's ordering mechanism, whereas the training-side heads use positive increments. On the final retained set, define

$$\begin{aligned} u_i = \hat{q}_i^{0.9} - \hat{q}_i^{0.5}, \qquad v_i = \hat{q}_i^{0.9} - \hat{q}_i^{0.1}, \\ d_i = [\hat{q}_i^{0.1}, \hat{q}_i^{0.5}, \hat{q}_i^{0.9}, v_i, u_i]. \end{aligned} \tag{5}$$

The gaps compare the predicted 90th percentile with the median and the 10th percentile on the log-residual scale. A projection $p(\cdot)$ and sigmoid gate $g(\cdot)$ produce a residual feature prompt,

$$\tilde{h}_i = h_i + \lambda_p g(d_i) \odot p(d_i). \tag{6}$$

We set $\lambda_p = 0.4$. The residual connection retains a direct path from the host features, while the learned gate scales the descriptor-driven update separately in each channel.

**Residual-sensitive weights and geometry.** The final head predicts mask logits $a_i$ and mass logits $b_i$. Spread adjusts both channels:

$$a_i' = a_i - \alpha_u u_i - \alpha_v v_i, \quad b_i' = b_i - \beta_u u_i - \beta_v v_i, \quad w_i = \frac{e^{b_i'} \sigma(a_i')}{\sum_{j \in \mathcal{I}} e^{b_j'} \sigma(a_j') + \eta}, \tag{7}$$

where $\sigma$ is the sigmoid and $\eta$ stabilizes normalization. Coefficients are given in Section 4. We use row-major vectorization $e = \mathrm{vec}_{\mathrm{row}}(E)$ and design row $A_i = (\mathbf{x}_i')^\top \otimes \mathbf{x}_i^\top$, so $A_i e = \mathbf{x}_i'^\top E \mathbf{x}_i$.

Weighted eight-point estimation (Hartley & Zisserman, 2004) solves

$$\hat{e} = \arg\min_{\|e\|_2=1} e^\top A^\top \operatorname{diag}(\mathbf{w}) A e, \tag{8}$$

and reshapes $\hat{e}$ with the same row-major convention. The gaps describe residual spread on the log scale. They enter the fitting rule directly, without a parametric density or conversion to variance. The explicit gap penalties have a monotone effect on relative weight under fixed logits.

For distinct $i, j$, holding the final head logits and other descriptor coordinates fixed gives

$$\frac{\partial \log(w_i/w_j)}{\partial u_i} = -\beta_u - \alpha_u[1 - \sigma(a_i')] \le 0, \tag{9}$$

with the analogous result for $v_i$. Nonnegative coefficients make relative weight non-increasing along either gap coordinate. This property concerns the explicit weighting path: the total response also includes changes in logits through feature conditioning. Common positive weight scaling leaves the ideal eigensolver unchanged.

**Joint correspondence and geometry supervision.** Training retains the host's progressive classification objective. Let $\mathcal{I}_\ell$ denote the correspondence indices at prediction interface $\ell$, and let $s_{\ell i}$ be its scalar logit. Labels and residuals are gathered with the same indices after each selection. For RQR-F, the host scales positive logits according to their distance from the inlier threshold:

$$\kappa_{\ell i} = y_i \exp\left(-\frac{|r_i - \epsilon_g|}{\epsilon_g}\right) + (1 - y_i), \qquad \pi_{\ell i} = \sigma(\kappa_{\ell i} s_{\ell i}). \tag{10}$$

This scaling is used only in the classification loss; geometric weights remain those of Eq. (7). Let $n_\ell^+$ and $n_\ell^-$ count the positive and negative labels in $\mathcal{I}_\ell$, and set $\omega_\ell = \max\{1, \max(n_\ell^-, 1)/\max(n_\ell^+, 1)\}$. The per-pair classification objective is

$$\mathcal{L}_{\text{cls}} = -\sum_\ell \frac{1}{|\mathcal{I}_\ell|} \sum_{i \in \mathcal{I}_\ell} \left[\omega_\ell y_i \log \pi_{\ell i} + (1 - y_i) \log(1 - \pi_{\ell i})\right]. \tag{11}$$

A stable logit-based implementation avoids taking logarithms of saturated probabilities. Positive-class balancing prevents the numerous outliers from dominating training; summing over interfaces also supervises the decisions made before the final fit.

Geometry supervision evaluates the predicted matrix on virtual correspondences $\mathcal{V}$ consistent with the ground-truth epipolar geometry. Using Eq. (1) with $\hat{E}$ in place of $E^\star$, we define

$$\mathcal{L}_{\text{geo}} = \frac{1}{|\mathcal{V}|} \sum_{j \in \mathcal{V}} \min\{r_j(\hat{E}), \gamma_e\}, \tag{12}$$

$$\mathcal{L}_{\text{full}} = \lambda_c \mathcal{L}_{\text{cls}} + \lambda_e(t) \mathcal{L}_{\text{geo}} + \lambda_q \mathcal{L}_q, \tag{13}$$

where $t$ is the training update, and losses are averaged across image pairs. The geometric term uses the final matrix; pinball supervision uses the final retained residuals. We set the clipping margin $\gamma_e$ to $0.1$, $\lambda_c$ to $1$, and $\lambda_q$ to $0.32$. The geometry coefficient is zero for the first 20,000 updates and $0.5$ thereafter, allowing the pruning scores to develop before eigensolver gradients are introduced. Quantile supervision is active from the start.

These objectives act at complementary levels. Classification separates inliers from outliers, pinball loss distinguishes residual levels within either class, and geometry supervision measures the consistency of their joint estimate. Gradients from the latter pass through the weighted solver to the feature prompt and weighting channels. Discrete retained indices select the supervised subset rather than defining a differentiable selection operation. Auxiliary integrations instead preserve each host's native losses: shared-feature RQR adds $\lambda_q \mathcal{L}_q$, while score guidance additionally updates native scores using the detached residual teacher. This distinction keeps the training-only integrations separate from RQR-F's inference feedback.

# 4 EXPERIMENTS

## 4.1 EXPERIMENTAL SETUP

**Datasets.** Models train on outdoor YFCC with SIFT and are tested on 4,000 held-out pairs (Thomee et al., 2016). Frozen transfer uses SuperPoint and LoFTR on YFCC, and all three features on indoor SUN3D (DeTone et al., 2018; Sun et al., 2021; Xiao et al., 2013): 14,872 pairs for SIFT/SuperPoint and 14,196 for LoFTR. HPatches provides 580 pairs, split into 285 illumination and 295 viewpoint cases (Balntas et al., 2017). Each native/RQR pair receives the same correspondence inputs. Pose inputs use camera-normalized coordinates; the RQR-F training configuration uses 2,000 SIFT keypoints per image. No target-domain adaptation or homography supervision is used.

**Evaluation metrics.** Pose AUC@5/10/20 integrates recall up to each angular tolerance, using the larger rotation/translation error. Homography AUC@3/5/10px uses mean image-corner error. AUCs are percentages; failures remain in the denominator. Paired comparisons use identical inputs with weighted fitting or RANSAC. Natural-outlier strata use ground-truth ratios and 300 pair-bootstrap resamples; overall solver comparisons use 10,000 paired resamples. These 95% intervals condition on fixed models and solver outputs. Table subscripts give AUC-point changes.

**Implementation details.** Paired backbones use 500,000 updates and batch size 32. Adam starts at $10^{-3}$, except $10^{-4}$ for DeMatch++/GeoMoE, with host decay schedules. Score guidance ramps to $0.05$ over updates 20,000–50,000 (Appendix A). Appendix E details the depth-20 RQR-F encoder. Its ablation uses 600,000 updates, batch size 16, and Adam at $10^{-3}$, with 100,000-update warmup and decay by $0.4$ at updates 200,000/400,000. In Eq. (7), $(\alpha_u, \alpha_v) = (0.12, 0.09)$, $(\beta_u, \beta_v) = (0.16, 0.10)$, and $\eta = 10^{-5}$. The linear warmup starts at $0.01$ times the base learning rate. RQR-F uses pinball weight $0.32$, residual stabilizer $10^{-6}$, and prompt scale $0.4$. Validation-selected checkpoints remain fixed across target features and datasets.

**Comparison methods.** PointCN, OANet, DeMatch++, and GeoMoE use score guidance; CLNet/BCLNet use shared-feature supervision, restricted to the final interface for BCLNet. RQR-F ablates quantile supervision and weighting, then prompting, on a fixed host. Median, Huber, and Laplace are continuous-target controls; SuperPoint/LoFTR supply input matches.

## 4.2 POSE AND HOMOGRAPHY ESTIMATION

**Residual learning across pruners.** Table 1 compares each native pruner with its RQR variant on identical test correspondences. Both retain the host inference pipeline, so the comparison tests residual supervision without replacing the encoder or geometric estimator. Weighted estimation and RANSAC reveal whether the benefit persists across solver choices.

Table 1: Paired pose estimation on YFCC with SIFT. AUC (%).

| Method | Weighted eight-point | | | RANSAC | | |
|---|---|---|---|---|---|---|
| | AUC@5 | AUC@10 | AUC@20 | AUC@5 | AUC@10 | AUC@20 |
| PointCN (Yi et al., 2018) | 9.03 | 23.08 | 43.12 | 24.71 | 41.56 | 58.64 |
| **PointCN + RQR** | $9.32_{+0.30}$ | $24.26_{+1.19}$ | $44.37_{+1.25}$ | $25.77_{+1.06}$ | $42.75_{+1.19}$ | $59.53_{+0.89}$ |
| OANet (Zhang et al., 2019) | 17.43 | 37.89 | 58.43 | 27.85 | 46.26 | 63.62 |
| **OANet + RQR** | $18.01_{+0.58}$ | $38.32_{+0.42}$ | $58.95_{+0.52}$ | $28.50_{+0.65}$ | $47.32_{+1.06}$ | $64.38_{+0.75}$ |
| CLNet (Zhao et al., 2021) | 27.27 | 47.34 | 66.16 | 31.55 | 51.38 | 68.81 |
| **CLNet + RQR** | $27.09_{-0.18}$ | $47.24_{-0.10}$ | $66.20_{+0.04}$ | $32.31_{+0.76}$ | $51.80_{+0.43}$ | $69.24_{+0.43}$ |
| BCLNet (Miao et al., 2024) | 33.36 | 55.00 | 72.42 | 34.39 | 55.13 | 72.07 |
| **BCLNet + RQR** | $37.13_{+3.77}$ | $57.80_{+2.79}$ | $73.81_{+1.39}$ | $35.19_{+0.81}$ | $55.14_{+0.02}$ | $71.99_{-0.07}$ |
| DeMatch++ (Zhang et al., 2025b) | 33.47 | 55.37 | 72.40 | 33.98 | 53.63 | 70.52 |
| **DeMatch++ + RQR** | $34.08_{+0.61}$ | $55.74_{+0.36}$ | $72.83_{+0.43}$ | $34.53_{+0.55}$ | $54.49_{+0.86}$ | $71.35_{+0.83}$ |
| GeoMoE (Le & Ma, 2026) | 33.58 | 56.02 | 73.48 | 34.54 | 54.55 | 71.19 |
| **GeoMoE + RQR** | $34.25_{+0.67}$ | $56.30_{+0.29}$ | $73.53_{+0.05}$ | $34.94_{+0.40}$ | $54.48_{-0.06}$ | $71.04_{-0.15}$ |

Weighted AUC increases at all tolerances for five hosts; CLNet instead gains in RANSAC while losing 0.18/0.10 weighted points at $5°/10°$. BCLNet's weighted AUC@5 gain is 3.77 points, with a paired 95% interval of [2.98,4.56]; its RANSAC gain is 0.81, with interval $[-0.10,1.70]$. Appendix B reports all hosts. RANSAC refits from filtered matches without continuous fitting weights. For CLNet/BCLNet, that filter itself depends on the predicted matrix. The results therefore show solver sensitivity, without isolating changes in score calibration or candidate membership as its cause.

**Generalization to homography estimation.** Table 2 evaluates epipolar-trained pruners on 580 HPatches pairs. Nearest-neighbor matching uses up to 4,000 SIFT keypoints per image, with coordinates centered and scaled by $\max(W-1, H-1)$. Both solvers filter at predicted epipolar residual $10^{-4}$; DLT weights by inverse residual, while RANSAC uses a 5px reprojection threshold. PointCN, OANet, DeMatch++, and GeoMoE improve all DLT AUCs. OANet gains in both illumination and viewpoint subsets, whereas DeMatch++ gains only under illumination. Mixed RANSAC changes show that improved soft fitting does not ensure a better sampled consensus.

Table 2: Homography estimation on HPatches. Corner-error AUC (%).

| Method | Weighted DLT | | | RANSAC | | |
|---|---|---|---|---|---|---|
| | AUC@3px | AUC@5px | AUC@10px | AUC@3px | AUC@5px | AUC@10px |
| PointCN (Yi et al., 2018) | 18.58 | 25.82 | 35.15 | 50.52 | 62.23 | 74.90 |
| **PointCN + RQR** | 19.95$_{+1.37}$ | 26.98$_{+1.16}$ | 36.42$_{+1.27}$ | 51.84$_{+1.32}$ | 63.46$_{+1.23}$ | 76.00$_{+1.10}$ |
| OANet (Zhang et al., 2019) | 19.81 | 27.92 | 38.19 | 52.16 | 63.62 | 76.19 |
| **OANet + RQR** | 22.82$_{+3.01}$ | 31.24$_{+3.32}$ | 41.71$_{+3.52}$ | 52.72$_{+0.56}$ | 64.43$_{+0.81}$ | 76.87$_{+0.68}$ |
| CLNet (Zhao et al., 2021) | 26.67 | 36.45 | 49.17 | 53.82 | 66.20 | 79.01 |
| **CLNet + RQR** | 26.85$_{+0.18}$ | 36.92$_{+0.47}$ | 48.74$_{-0.43}$ | 52.89$_{-0.93}$ | 65.03$_{-1.17}$ | 78.12$_{-0.89}$ |
| BCLNet (Miao et al., 2024) | 25.16 | 35.57 | 48.35 | 52.92 | 65.46 | 78.29 |
| **BCLNet + RQR** | 25.98$_{+0.82}$ | 34.58$_{-0.99}$ | 46.07$_{-2.28}$ | 54.04$_{+1.12}$ | 65.92$_{+0.46}$ | 78.49$_{+0.20}$ |
| DeMatch++ (Zhang et al., 2025b) | 23.04 | 31.92 | 43.22 | 53.77 | 65.33 | 77.82 |
| **DeMatch++ + RQR** | 23.60$_{+0.56}$ | 32.83$_{+0.91}$ | 43.41$_{+0.19}$ | 54.03$_{+0.26}$ | 65.82$_{+0.49}$ | 78.01$_{+0.19}$ |
| GeoMoE (Le & Ma, 2026) | 21.44 | 29.93 | 41.09 | 52.86 | 64.71 | 77.05 |
| **GeoMoE + RQR** | 23.55$_{+2.11}$ | 31.72$_{+1.79}$ | 41.94$_{+0.85}$ | 52.64$_{-0.22}$ | 64.08$_{-0.63}$ | 76.21$_{-0.84}$ |

### 4.3 Robustness and Cross-Domain Generalization

**Robustness under natural outliers.** Figure 3 compares outdoor YFCC and indoor SUN3D pairs with SIFT in four outlier strata. The fixed models are evaluated across bins to test whether aggregate gains extend to sparse geometric support.

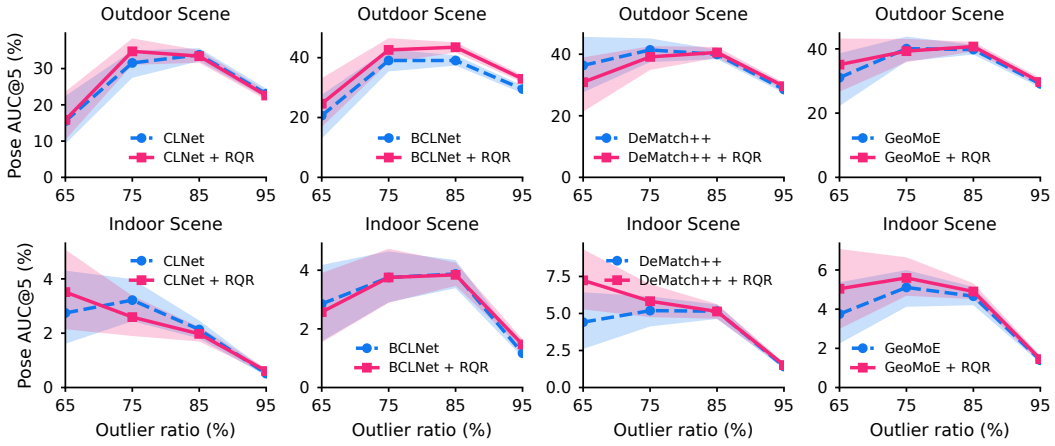



Figure 3: **Natural-outlier robustness.** Weighted pose AUC@5 with 95% pair-bootstrap intervals.

BCLNet improves outdoor AUC@5 in every bin, including a 3.41-point gain above 90% outliers. DeMatch++ gains 1.06 points in this hardest bin despite losses in the two lower-outlier bins. Its aggregate benefit therefore reflects better behavior where reliable support is sparse, rather than a uniform upward shift. GeoMoE gains 0.58 points in the same outdoor bin and improves across all indoor bins, although its indoor absolute gains decrease as contamination rises.

CLNet gains 3.20 points in the outdoor $(70, 80]$ bin but loses 0.50 above 90% outliers, showing that shared-feature supervision helps some contamination regimes more than others. The lowest outdoor bin contains 52 pairs, versus 2,262 in the highest; bootstrap intervals quantify test-pair variability conditional on the trained models, not variability across training runs. Because contamination and scene difficulty vary together, these are observational robustness comparisons.

**Changing local features and scenes.** Figure 4 evaluates five settings without adaptation. On YFCC with LoFTR, the four displayed hosts improve weighted AUC@20; GeoMoE gains 5.20 points and DeMatch++ gains 4.12. The trend is not universal: PointCN and OANet lose 2.14 and 2.52 points, respectively, on the same feature setting.

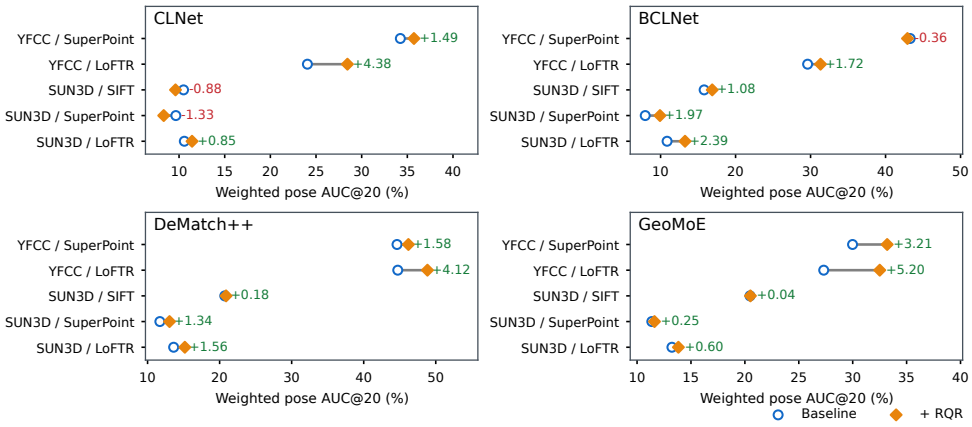



Figure 4: **Cross-domain and cross-feature generalization.** Weighted AUC@20; annotations give the RQR change in points.

SUN3D additionally changes scene geometry. DeMatch++ and GeoMoE improve with all indoor feature types; BCLNet gains 2.39 points with LoFTR and 1.08 with SIFT. CLNet improves with LoFTR but declines with SIFT and SuperPoint. Changing features alters the coordinates and residuals presented to a SIFT-trained pruner. Opposite RQR effects on the same LoFTR inputs indicate that transfer depends on how the host uses residual supervision. Match density and context aggregation are plausible factors, but separating their contributions requires controlled comparisons. These results establish host-dependent transfer without assigning the divergence to encoder capacity alone.

### 4.4 Ablation Study

**Inference components.** Table 3 holds the Transformer host and training schedule fixed. Adding quantile prediction, supervision, and weighting improves weighted AUC@5/10 by 1.27/0.93 points. These additions are evaluated jointly. Each configuration runs for 600,000 updates, with validation every 10,000 updates and its own selected checkpoint. The gated prompt adds a further 0.25/0.14 points. Full RQR-F adds 18,947 parameters (0.19%) and 0.89 ms to the common host's 88.93 ms synchronized forward on one fixed 2,000-match input on an RTX 3090 (Appendix D). The prompt's additional gain suggests that residual information can assist feature refinement beyond its joint use in supervision and weighting.

Table 3: Ablation Study.

| Base | RQR | Prompt | Weighted AUC (%) | |
|---|---|---|---|---|
| | | | AUC@5 | AUC@10 |
| ✓ | | | 37.63 | 59.05 |
| ✓ | ✓ | | $38.90_{+1.27}$ | $59.98_{+0.93}$ |
| ✓ | ✓ | ✓ | $\mathbf{39.15}_{+1.52}$ | $\mathbf{60.12}_{+1.07}$ |

**Residual targets and supervision placement.** Table 4 evaluates residual targets and supervision placement on BCLNet. Every continuous target improves over the native model. In the common all-interface search, Huber exceeds three-quantile RQR by 2.71/1.83/0.96 AUC points; Laplace also improves on it at all tolerances. Continuous residual supervision therefore accounts for part of the gain.

Table 4: Residual targets and supervision placement.

| Target | Weighted AUC (%) | | |
|---|---|---|---|
| | AUC@5 | AUC@10 | AUC@20 |
| None (baseline) | 33.36 | 55.00 | 72.42 |
| *All prediction interfaces* | | | |
| Median (three outputs) | 34.26 | 56.10 | 73.21 |
| Huber† | **36.30** | **57.17** | **73.77** |
| Laplace† | 35.33 | 56.98 | 73.76 |
| RQR, 3q† | 33.59 | 55.34 | 72.81 |
| RQR, 3q (fixed) | 34.80 | 56.13 | 73.17 |
| RQR, 5q (fixed) | 35.58 | 56.44 | 73.32 |
| *Final prediction interface* | | | |
| RQR, 3q (fixed) | **37.13** | **57.80** | **73.81** |

Entries marked † select $\lambda_q$ from $\{0.1, 0.25, 0.5\}$ on validation data; fixed-weight rows use $0.25$. At this fixed weight, moving three-quantile supervision from all interfaces to the final interface improves AUC by 2.33/1.67/0.64 points. This comparison identifies a placement effect within RQR; final-interface Huber and Laplace remain untested. RQR-F instead tests using the descriptor during inference on a fixed host. A single point estimate does not supply its quantile gaps; a parametric distribution could provide alternative spread descriptors. Table 3 supports the proposed feedback path, while a matched comparison against such alternatives remains open.

Held-out diagnostics separate residual prediction from calibration: nominal 80% intervals cover 77.66–80.42% for CLNet's three interfaces and 75.74% for RQR-F (Appendix C). Thus the observed pose gains do not imply calibrated uncertainty.

### 4.5 Discussion

Figure 5 compares the selected matches of OANet and RQR-F. Some green matches appear visually implausible because the label tests epipolar consistency rather than physical point identity. An incorrect correspondence near the epipolar line can pass the residual threshold. This is a limitation of geometric inlier labels; the examples do not establish the prevalence of annotation errors.

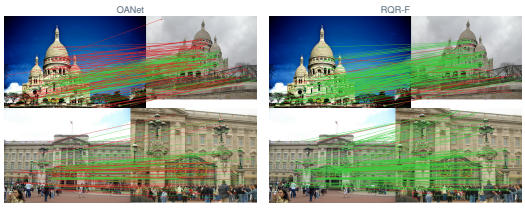



Figure 5: OANet and RQR-F on YFCC. Green/red: geometric inliers/outliers.

RQR inherits this ambiguity: low residual does not certify a true match, and pose noise affects the supervision. Appearance and multi-view constraints could complement geometric labels. Epipolar gaps are not pixel reprojection variances, so using them in bundle adjustment would require a separate model and evaluation. The qualitative comparison uses different encoders; the fixed-host ablation assesses the added inference components.

## 5 Conclusion

RQR represents geometric residual location and tail spread through ordered quantile targets, adding information that binary inlier supervision leaves unspecified. Training-time integrations improve all weighted pose AUCs for five of six tested pruners. RQR-F uses the descriptor during inference to refine features and fitting weights, with gains in the fixed-host ablation. Continuous-target controls establish the value of residual supervision and the importance of its placement; paired solver and frozen-transfer evaluations characterize how those gains vary with the estimator, host, and inputs. Quantile gaps summarize predicted spread, while held-out coverage measures their calibration separately from pose accuracy. Matched comparisons of alternative descriptors and supervision locations would clarify the residual information needed at each interface. Appearance and multi-view constraints offer complementary evidence for distinguishing geometric consistency from physical correspondence identity.

### Reproducibility Statement

The mathematical formulations, residual loss objectives, and inference operations are specified in Section 3. Section 4 and Appendices A and E detail the datasets and evaluation splits, training schedules, hyperparameters, and solver configurations. All paired comparisons use identical correspondence inputs and fixed, validation-selected checkpoints. Pair-bootstrap resampling characterizes test-pair variation conditional on these trained models, rather than variation across training runs. Complete training and evaluation code, environment configurations, and pretrained model weights will be released publicly upon publication.

### Ethics Statement

This work studies geometric correspondence pruning using public academic benchmarks (YFCC, SUN3D, and HPatches), without recruiting human participants or collecting new images. Correspondence estimation supports 3D reconstruction and visual localization, but geometric inlier labels reflect geometric residual consistency rather than certified physical feature identity. Downstream safety-critical or privacy-sensitive systems should account for this distinction when relying on predicted residual descriptors.

### AI Usage Statement

Generative AI (ChatGPT) was used solely for code debugging, language polishing, and assistance with formatting and table editing. It was not used to formulate scientific hypotheses, develop the proposed methodology, conduct experiments, or generate scientific conclusions. All AI-assisted outputs were carefully reviewed and, where necessary, revised by the authors. The authors take full responsibility for the content of the manuscript.

## References

Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5173–5182, 2017.

JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng. GMS: Grid-Based Motion Statistics for Fast, Ultra-Robust Feature Correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4181–4190, 2017.

Eric Brachmann and Carsten Rother. Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses. In *IEEE/CVF International Conference on Computer Vision*, pp. 4322–4331, 2019.

Peiqi Chen, Lei Yu, Yi Wan, Yingying Pei, Xinyi Liu, Yongxiang Yao, Yingying Zhang, Lixiang Ru, Liheng Zhong, Jingdong Chen, Ming Yang, and Yongjun Zhang. CasP: Improving Semi-Dense Feature Matching Pipeline Leveraging Cascaded Correspondence Priors for Guidance. In *IEEE/CVF International Conference on Computer Vision*, pp. 28063–28072, 2025.

Ondrej Chum, Jiri Matas, and Josef Kittler. Locally Optimized RANSAC. In *DAGM Symposium on Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pp. 236–243. Springer, 2003. doi: 10.1007/978-3-540-45243-0_31.

Luanyuan Dai, Yizhang Liu, Jiayi Ma, Lifang Wei, Taotao Lai, Changcai Yang, and Riqing Chen. MS2DG-Net: Progressive Correspondence Learning via Multiple Sparse Semantics Dynamic Graph. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8973–8982, 2022.

Luanyuan Dai, Xiaoyu Du, and Jinhui Tang. LeCoT: Revisiting Network Architecture for Two-View Correspondence Pruning. *Science China Information Sciences*, 69(1):112104, 2026. doi: 10.1007/s11432-024-4555-x.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 224–236, 2018.

Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981. doi: 10.1145/358669.358692.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004. doi: 10.1017/cbo9780511811685.

Roger Koenker and Gilbert Bassett, Jr. Regression Quantiles. *Econometrica*, 46(1):33–50, 1978. doi: 10.2307/1913643.

Jiajun Le and Jiayi Ma. GeoMoE: Divide-and-Conquer Motion Field Modeling with Mixture-of-Experts for Two-View Geometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5845–5853, 2026. doi: 10.1609/aaai.v40i7.37506.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A Framework for Attention-Based Permutation-Invariant Neural Networks. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 2019.

Shuyuan Lin, Mengtin Lo, Haosheng Chen, Yanjie Liang, and Qiangqiang Wu. MGCA-Net: Multi-Graph Contextual Attention Network for Two-View Correspondence Learning. In *International Joint Conference on Artificial Intelligence*, pp. 1539–1547, 2025. doi: 10.24963/ijcai.2025/172.

Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *IEEE/CVF International Conference on Computer Vision*, pp. 17627–17638, 2023.

Yizhang Liu, Yanping Li, and Shengjie Zhao. TransMatch: Transformer-Based Correspondence Pruning via Local and Global Consensus. *Pattern Recognition*, 159:111120, 2025. doi: 10.1016/j.patcog.2024.111120.

David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. doi: 10.1023/b:visi.0000029664.99615.94.

Yifan Lu, Jiajun Le, Zizhuo Li, Yixuan Yuan, and Jiayi Ma. DeMo: Deep Motion Field Consensus with Learnable Kernels for Two-View Correspondence Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5829–5837, 2025. doi: 10.1609/aaai.v39i6.32622.

Xiangyang Miao, Guobao Xiao, Shiping Wang, and Jun Yu. BCLNet: Bilateral Consensus Learning for Two-View Correspondence Pruning. In *AAAI Conference on Artificial Intelligence*, volume 38, pp. 4225–4232, 2024. doi: 10.1609/aaai.v38i5.28218.

Dror Moran, Yuval Margalit, Guy Trostianetsky, Fadi Khatib, Meirav Galun, and Ronen Basri. Consensus Learning with Deep Sets for Essential Matrix Estimation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 101529–101551, 2024. doi: 10.52202/079017-3220.

René Ranftl and Vladlen Koltun. Deep Fundamental Matrix Estimation. In *European Conference on Computer Vision*, pp. 284–299, 2018.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4938–4947, 2020.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching with Transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8922–8931, 2021.

Weiwei Sun, Wei Jiang, Eduard Trulls, Andrea Tagliasacchi, and Kwang Moo Yi. ACNe: Attentive Context Normalization for Robust Permutation-Equivariant Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11286–11295, 2020.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The New Data in Multimedia Research. *Communications of the ACM*, 59(2):64–73, 2016. doi: 10.1145/2812802.

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.

Jianxiong Xiao, Andrew Owens, and Antonio Torralba. SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels. In *IEEE International Conference on Computer Vision*, pp. 1625–1632, 2013.

Fei Xue, Sven Elflein, Laura Leal-Taixé, and Qunjie Zhou. MATCHA: Towards Matching Anything. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27081–27091, 2025.

Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to Find Good Correspondences. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2666–2674, 2018.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3391–3401, 2017.

Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning Two-View Correspondences and Geometry Using Order-Aware Network. In *IEEE/CVF International Conference on Computer Vision*, pp. 5845–5854, 2019.

Jintao Zhang, Zimin Xia, Mingyue Dong, Shuhan Shen, Linwei Yue, and Xianwei Zheng. CoMatcher: Multi-View Collaborative Feature Matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21970–21980, 2025a.

Lily Zhang, Veronica Tozzo, John Higgins, and Rajesh Ranganath. Set Norm and Equivariant Skip Connections: Putting the Deep in Deep Sets. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 26559–26574. PMLR, 2022.

Shihua Zhang and Jiayi Ma. ConvMatch: Rethinking Network Design for Two-View Correspondence Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2920–2935, 2024. doi: 10.1109/tpami.2023.3334515.

Shihua Zhang, Zizhuo Li, and Jiayi Ma. DeMatch++: Two-View Correspondence Learning via Deep Motion Field Decomposition and Respective Local-Context Aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11234–11251, 2025b. doi: 10.1109/tpami.2025.3596598.

Chen Zhao, Yixiao Ge, Feng Zhu, Rui Zhao, Hongsheng Li, and Mathieu Salzmann. Progressive Correspondence Pruning by Consensus Learning. In *IEEE/CVF International Conference on Computer Vision*, pp. 6464–6473, 2021.

Zhen Zhong, Guobao Xiao, Linxin Zheng, Yan Lu, and Jiayi Ma. T-Net: Effective Permutation-Equivariant Network for Two-View Correspondence Learning. In *IEEE/CVF International Conference on Computer Vision*, pp. 1950–1959, 2021.

# A Training and Evaluation Protocols

**Evaluation denominators.** All comparisons within a domain use the same image pairs. A failed estimate remains in the AUC denominator rather than being removed. Pose error is the larger of rotation and translation angular errors; homography error measures image-corner displacement in pixels. These tolerances refer to different geometric tasks. Frozen transfer uses no adaptation on the target domain. Each native and RQR model uses its own checkpoint. Paired models receive identical feature inputs, and each checkpoint remains fixed across target domains. Results describe one trained instance per configuration; pair-bootstrap intervals characterize test-pair variation.

**Score-guidance objectives and gradient paths.** Let $\bar{s}_i$ and $\bar{R}_i$ denote standardized native scores and risks, using per-pair standard deviations floored at $10^{-4}$, and let $N$ be the block's correspondence count. The additional objective is averaged over prediction blocks:

$$\mathcal{L}_{\text{aux}} = \lambda_Q \mathcal{L}_q + \lambda_B \mathcal{L}_{\text{boundary}} + \lambda_G \mathcal{L}_{\text{geometry-rank}} + \gamma(t)\mathcal{L}_{\text{guide}}, \tag{14}$$

where $\lambda_Q = \lambda_B = 0.1$, $\lambda_G = 0.1$ for DeMatch++/GeoMoE and zero otherwise, and $\gamma(t) = 0.05 \min\{1, \max\{0, (t - 20000)/30000\}\}$. Pinball supervision uses $\log \max(r_i, 10^{-8})$. PointCN, DeMatch++, and GeoMoE multiply the square-root residual risk by the eight-point design-row norm before standardization; OANet omits this factor.

The boundary loss averages $\text{softplus}(-\bar{R}_n + \bar{R}_p + 0.2)$ over ground-truth outlier–inlier pairs $(n, p)$. OANet/PointCN use a band of $\max\{4, \text{round}(0.1N)\}$ ranks on each side of the top-10% boundary in teacher-risk order, capped at 64 candidates per class. DeMatch++/GeoMoE instead gather native logits nearest zero from either side, then form class-based pairs.

OANet aligns $\bar{s}_i$ with detached $-\bar{R}_i$ using Smooth L1, weighted by a detached gate $\exp(-|\bar{s}_i - b|/0.5)$, where $b$ is its native top-10% cutoff; the denominator is the gate sum over the batch, floored at one. The other hosts use

$$\mathcal{L}_{\text{guide}} = \zeta \left\langle \text{softplus}\left( \frac{m - (\bar{s}_j - \bar{s}_i)}{T_r} \right) \right\rangle_{\bar{R}_j + \Delta_R < \bar{R}_i}, \tag{15}$$

where $\zeta = 0.125$, $m = 0.1$, $T_r = \Delta_R = 0.25$, and $i$ and $j$ are selected and rejected candidates near the native boundary: top-10% for PointCN, zero for DeMatch++/GeoMoE. Each side contains at most $\min\{64, \max\{1, \text{round}(0.1N)\}\}$ candidates, subject to availability.

DeMatch++/GeoMoE apply the same standardization to the target $\|A_i\|_2 \sqrt{r_i}$, obtaining $G_i$, and minimize

$$\mathcal{L}_{\text{geometry-rank}} = \left\langle \text{softplus}\left( \frac{m - \text{sgn}(G_i - G_j)(\bar{R}_i - \bar{R}_j)}{T_r} \right) \right\rangle_{|G_i - G_j| \geq \Delta_G}, \tag{16}$$

where $\Delta_G = 0.25$. Pairwise losses average valid pairs within each image pair, then average image pairs containing valid pairs; the loss is zero if none remain. Teacher feature/score inputs are detached. Guidance detaches predicted risks and updates native scores. Geometry ranking differentiates predicted risks, with detached targets and discrete candidate indices; native-score arguments select candidates but provide no differentiable loss path. Inference retains the native scores and solver.

# B Paired Solver Diagnostics

Table B.1 reports RQR-minus-native AUC changes on the same 4,000 YFCC pairs with SIFT. We draw 10,000 bootstrap samples of image-pair indices, applying each sample jointly to the native and RQR errors for both estimators. AUC is recomputed by the original empirical-CDF trapezoidal rule; percentile intervals use the 2.5th and 97.5th percentiles. Checkpoints and solver outputs remain fixed during resampling. These intervals describe test-pair variability conditional on the fitted models; they do not quantify training variability. Comparisons are paired within each host, with matched evaluation settings.

BCLNet's weighted gains have intervals above zero at all tolerances, whereas its RANSAC intervals contain zero. The weighted-minus-RANSAC gain differences are 2.96 [1.96,4.00], 2.78 [1.90,3.67], and 1.46 [0.83,2.11] points at $5^\circ/10^\circ/20^\circ$. The corresponding difference intervals include zero for

Table B.1: Paired pose AUC changes and 95% image-pair bootstrap intervals.

| Method | Estimator | RQR − native AUC (percentage points) | | |
|---|---|---|---|---|
| | | 5° | 10° | 20° |
| PointCN | Weighted | $+0.30$ $[-0.26, +0.86]$ | $+1.19$ $[+0.50, +1.87]$ | $+1.25$ $[+0.64, +1.88]$ |
| | RANSAC | $+1.06$ $[+0.22, +1.91]$ | $+1.19$ $[+0.34, +2.06]$ | $+0.89$ $[+0.13, +1.66]$ |
| OANet | Weighted | $+0.58$ $[-0.09, +1.27]$ | $+0.42$ $[-0.31, +1.16]$ | $+0.52$ $[-0.07, +1.12]$ |
| | RANSAC | $+0.65$ $[-0.24, +1.52]$ | $+1.06$ $[+0.21, +1.90]$ | $+0.75$ $[+0.02, +1.49]$ |
| CLNet | Weighted | $-0.18$ $[-0.96, +0.62]$ | $-0.10$ $[-0.87, +0.66]$ | $+0.04$ $[-0.56, +0.65]$ |
| | RANSAC | $+0.76$ $[-0.13, +1.65]$ | $+0.43$ $[-0.40, +1.28]$ | $+0.43$ $[-0.24, +1.08]$ |
| BCLNet | Weighted | $+3.77$ $[+2.98, +4.56]$ | $+2.79$ $[+2.11, +3.47]$ | $+1.39$ $[+0.89, +1.88]$ |
| | RANSAC | $+0.81$ $[-0.10, +1.70]$ | $+0.02$ $[-0.79, +0.81]$ | $-0.07$ $[-0.68, +0.54]$ |
| DeMatch++ | Weighted | $+0.61$ $[-0.11, +1.32]$ | $+0.36$ $[-0.25, +0.97]$ | $+0.43$ $[-0.04, +0.89]$ |
| | RANSAC | $+0.55$ $[-0.34, +1.40]$ | $+0.86$ $[+0.05, +1.65]$ | $+0.83$ $[+0.19, +1.48]$ |
| GeoMoE | Weighted | $+0.67$ $[-0.07, +1.40]$ | $+0.29$ $[-0.32, +0.89]$ | $+0.05$ $[-0.38, +0.48]$ |
| | RANSAC | $+0.40$ $[-0.49, +1.27]$ | $-0.06$ $[-0.86, +0.73]$ | $-0.15$ $[-0.77, +0.48]$ |

the other five hosts. For OANet, DeMatch++, GeoMoE, and CLNet, all three weighted-gain intervals include zero. This distinguishes a large measured BCLNet effect from smaller point estimates without interpreting inconclusive intervals as evidence of no effect.

**How the estimators use network outputs.** PointCN, OANet, DeMatch++, and GeoMoE pass matches with nonnegative final logits to RANSAC. CLNet and BCLNet instead compute epipolar residuals under the network-predicted matrix and retain values strictly below the smaller of $10^{-4}$ and the largest residual. RANSAC estimates a new essential matrix from these coordinates with a normalized-coordinate threshold of 0.001, without continuous network weights or sampling probabilities. Thus the CLNet/BCLNet candidate set still depends indirectly on weighted estimation. Weighted evaluation uses the network's matrix and approximately the best 10% of matches for pose decomposition; this subset is used for cheirality resolution, not to refit the weighted matrix. The two estimators consume the learned scores differently, so neither a gain nor a decline in one identifies the cause of a change in the other.

## C Held-out Residual Coverage

We evaluate the frozen validation-selected CLNet and full RQR-F checkpoints on all 4,000 held-out YFCC pairs with SIFT. No calibration map is fitted and no coefficient is chosen using these diagnostics. For each prediction interface, coverage is the fraction of retained correspondences satisfying $z_i \le \hat{q}_i^{\tau}$. The central interval checks $\hat{q}_i^{0.1} \le z_i \le \hat{q}_i^{0.9}$. Correspondence counts, rather than image pairs, define these fractions; 2,000 image-pair bootstrap resamples provide percentile 95% intervals. Figure C.1 and Table C.1 report all measured interfaces.

Table C.1: Held-out marginal coverage and transformed interval width.

| Model / interface | Count | Coverage (%) | | | | Width |
|---|---|---|---|---|---|---|
| | | $q^{0.1}$ | $q^{0.5}$ | $q^{0.9}$ | $[q^{0.1}, q^{0.9}]$ | |
| CLNet, interface 0 | 7,956,322 | 12.63 | 55.17 | 92.93 | 80.30 | 2.341 |
| CLNet, interface 1 | 3,977,964 | 11.82 | 50.58 | 89.49 | 77.66 | 2.506 |
| CLNet, interface 2 | 1,988,930 | 8.94 | 49.01 | 89.36 | 80.42 | 2.622 |
| RQR-F, final interface | 1,988,930 | 11.60 | 51.29 | 87.34 | 75.74 | 3.070 |

For CLNet, the three interfaces have nominal 80% interval coverages of 80.30%, 77.66%, and 80.42%. The intermediate interface undercovers, and the first interface's median coverage is 55.17%. Near-nominal central interval coverage can therefore coexist with biased individual quantiles. Each in-

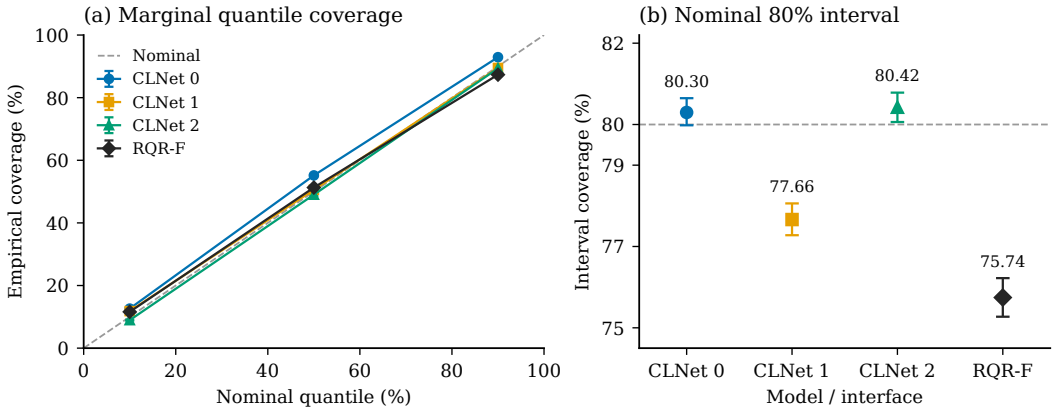



Figure C.1: Frozen-model coverage on YFCC with SIFT. Error bars: 95% image-pair bootstrap intervals.

terface retains a different population of correspondences, so these values do not measure a calibration trajectory on one unchanged set.

RQR-F has marginal coverages of 11.60%, 51.29%, and 87.34% at nominal levels 10%, 50%, and 90%. Its central interval covers 75.74%, below the nominal 80%. This is a measurable calibration limitation of the jointly trained descriptor, even though Table 3 records a pose improvement. The diagnostics do not isolate whether finite capacity, optimization, or the geometric objective causes the undercoverage.

The targets remain specific to each integration: CLNet uses $\log(1 + r/10^{-4})$, whereas RQR-F uses $\log(r + 10^{-6})$. Mean widths are measured on these respective transformed scales and should not be compared as common-unit residual uncertainties. Even exact pooled marginal coverage would not establish conditional coverage for individual correspondences or stability under a feature/domain shift. We use the outputs as learned residual descriptors, without a risk-controlled rejection or calibrated-variance claim.

## D Computational Cost on a Common Host

We measure the three configurations of Table 3 on the same RTX 3090, using one fixed real YFCC test pair with 2,000 SIFT correspondences, batch size one, FP32, and evaluation mode without gradients. Each of three rotated execution orders contains 20 warmup and 100 measured forwards per configuration. Table D.1 reports the mean and sample standard deviation over the resulting 300 measurements per configuration. CUDA synchronization brackets each forward.

Table D.1: Matched inference cost for the RQR-F component study.

| Configuration | Registered parameters | Active parameters | Forward (ms) mean ± SD | Peak (MiB) allocated |
|---|---|---|---|---|
| Common host | 10,149,862 | 10,149,862 | $88.93 \pm 0.66$ | 173.40 |
| Head + weighting | 10,168,809 | 10,150,249 | $89.43 \pm 0.73$ | 173.47 |
| Full RQR-F | 10,168,809 | 10,168,809 | $89.82 \pm 0.59$ | 173.47 |

The full feedback path adds 18,947 active parameters (0.187%) and 0.885 ms (1.00%) to the common host in this measurement. The quantile head contains 387 parameters; its feature projection and gate add 18,560. The head-only configuration registers the latter layers but does not execute them, explaining the distinction between registered and active parameters. Attention, pooling, and the expanded encoder belong to all three configurations and are not RQR overhead.

Timing includes the native CPU eigensolver and its device transfers, but excludes data loading, loss evaluation, pose recovery, and RANSAC. Peak allocated memory excludes the CUDA context and reserved cache. These timings measure model inference on a fixed input, excluding the remaining

pose-estimation pipeline. Training-only RQR heads can be removed because they do not feed the native inference outputs; that architectural property is distinct from this RQR-F measurement.

## E RQR-F Architecture and Implementation

Figure E.1 makes explicit the data flow of the inference-time model in Section 3.4. The two encoders share a design but have separate parameters. Residual prediction operates only on the final retained set; its outputs condition both correspondence features and fitting weights.

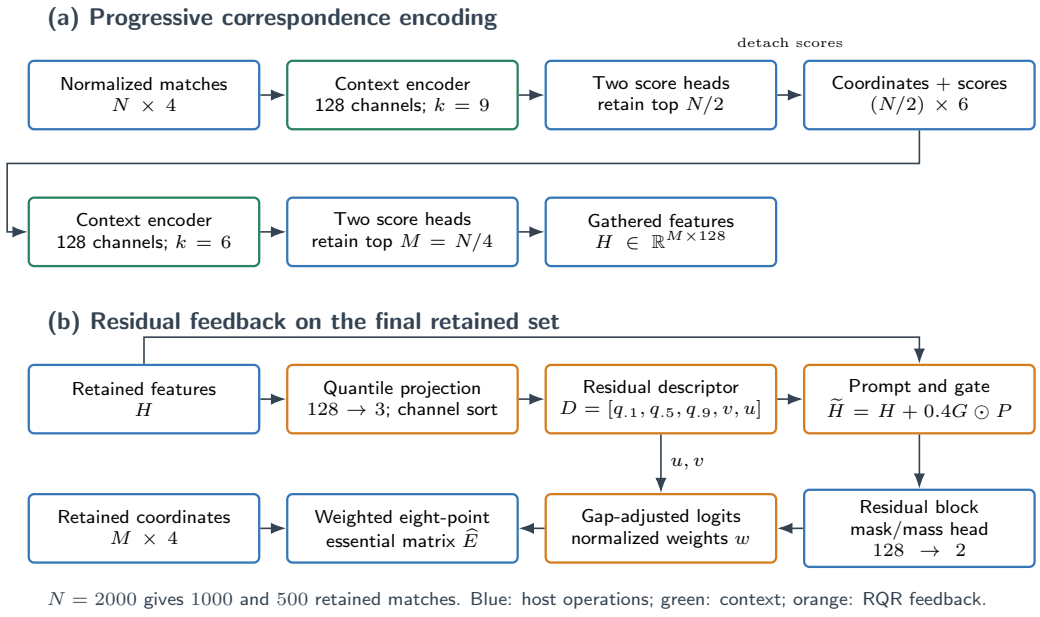



Figure E.1: RQR-F data flow and residual feedback. Dimensions omit the batch axis.

**Progressive encoding and selection.** The input has shape $B \times 1 \times N \times 4$, containing two camera-normalized image coordinates per match. The first encoder maps four channels to 128, predicts two scalar score vectors, and ranks by the second. It keeps the highest-scoring half and gathers both score vectors at those indices. Applying $\max(0, \tanh(s))$ to each gathered score produces two additional input channels. These scores are detached before concatenation with the retained coordinates. The second encoder therefore receives $B \times 1 \times (N/2) \times 6$. It repeats score prediction and half-set selection, gathering coordinates, residual labels, and 128-channel features with the same indices. For $N = 2000$, the two selections retain 1000 and 500 matches.

**Encoder depth and context.** Each encoder starts with a $1 \times 1$ convolution, batch normalization, ReLU, and five residual channel-mixing blocks. Two successive context units combine local attention and order-aware aggregation; scalar heads follow the first unit and the second unit's residual tail. For the depth-20 configuration, each context unit contains three local-attention blocks and three order-aware filters. Concatenation doubles the channel count to 256; a residual projection restores 128 channels, a reliability gate reweights them, and five residual blocks refine the result. A one-block tail precedes the second score head. The graph neighborhood size is nine in the first encoder and six in the second; both use 180 latent clusters. The depth setting controls these block counts rather than denoting a literal total of 20 convolutions.

**Final prediction path.** A $128 \to 3$ convolution predicts quantiles on the 500 retained feature vectors, with sorting along the quantile channel. The five-channel descriptor in Eq. (5) drives the prompt and gate. After the update in Eq. (6), one residual block and a $128 \to 2$ convolution produce mask and mass logits. The original predicted quantiles also supply the two gap penalties in Eq. (7); they are not recomputed from the prompted features. The weighted solver uses the

coordinates gathered at the second selection. The implemented RQR-F path contains no routing branch. Classification, geometry, and pinball losses have coefficients $1.0$, $0.5$, and $0.32$, respectively, with the geometry term activated after 20,000 updates.

### E.1 Internal blocks and gradient paths

Figure E.2 expands the context and residual-feedback operations. All feature convolutions shown are pointwise; neighborhood interaction is introduced by graph aggregation and latent pooling.

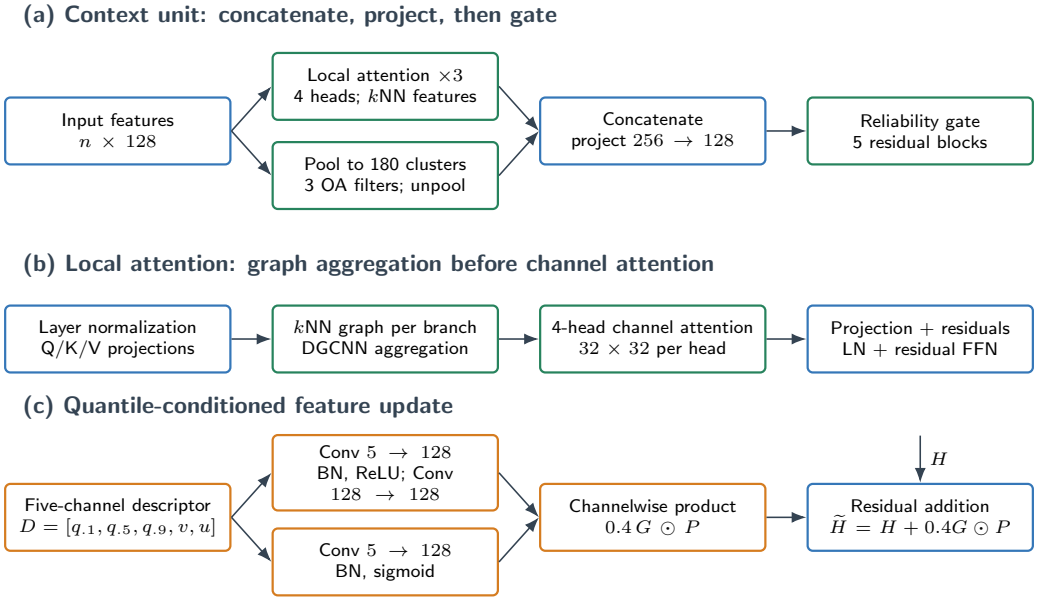



Figure E.2: Context fusion, local attention, and quantile-conditioned feature updates.

**Local and order-aware context.** The attention path applies layer normalization and separate query, key, and value projections, followed by feature-space $k$NN graph construction and DGCNN aggregation in each branch. The 128 channels are divided into four heads. Queries and keys are normalized across the correspondence axis; their product forms a $32 \times 32$ channel-attention matrix per head, scaled by a learned temperature and normalized by softmax. Thus local graph aggregation supplies match-neighborhood information before channel attention. Output projection and residual feed-forward operations complete the block. In parallel, differentiable pooling maps features to 180 clusters, three order-aware filters mix latent context, and unpooling returns it to the original match order. The two paths are concatenated before the residual projection; they are not directly added.

**Reliability and residual channel mixing.** The reliability gate adds a global-average branch to a local graph branch and applies a sigmoid. The global branch uses a $128 \to 32 \to 128$ bottleneck; the local branch uses graph aggregation at the encoder's neighborhood size. The resulting channelwise weights multiply the fused features. Residual channel-mixing blocks use two convolutions with instance and batch normalization, an intermediate ReLU, and a final ReLU after residual addition. A projection shortcut is used when the input and output widths differ.

**Feedback and supervision.** The prompt is $\mathrm{Conv}(5, 128)$–BN–ReLU–$\mathrm{Conv}(128, 128)$, and the gate is $\mathrm{Conv}(5, 128)$–BN–sigmoid. Their product is scaled by $0.4$ and added to the retained features. Pinball supervision uses the matched log-residual targets gathered through both selections. The logarithm compresses large residuals while preserving their order; the additive stabilizer controls behavior near zero. Gradients from geometry pass through the final logits and feedback branches; the direct gap penalties also provide a path to the quantile head. Channel sorting routes gradients

to the selected quantile values almost everywhere; the discrete top-half indices have no ranking gradient. The first encoder receives its own classification supervision, while the detached score channels prevent downstream losses from backpropagating through those inputs. Training schedules, loss coefficients, and solver stabilization are given in Section 4; Appendix D reports the separately measured cost of these additions.