BRIDGEMATCH: CONDITIONAL TRANSPORT BRIDGES IN MATCHING MATRIX SPACE FOR 3D DEFORMABLE REGISTRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable matching between partially observed, deforming point clouds requires global context and fine geometric detail. Coarse candidate selection can exclude correct fine-level correspondences. We present BridgeMatch, a unified conditional transport framework with the matching matrix itself as the evolving state. Coarse diffusion establishes global matching hypotheses; hierarchy-preserving lifting expands them into a structured high-resolution source. Geometry-conditioned ODE and SDE bridges continue refinement in the complete fine-level candidate space, allowing coarse errors to be corrected. The deterministic endpoint-parameterized conditional flow matching (CFM) design improves matching through iteration, while the SDE forward drift enables effective few-step refinement. Experiments on 4DMatch and 4DLoMatch demonstrate competitive matching and non-rigid registration, with transfer to CAPE and DeepDeform without target-domain training.

1 INTRODUCTION

Non-rigid point cloud registration estimates correspondences and spatially varying deformations between point clouds. Applications include dynamic reconstruction Prokudin et al. (2023), medical imaging Zou et al. (2022); Yang et al. (2025), and deformable-object manipulation Shi et al. (2022); Lin et al. (2023). Partial visibility, outliers, and large motion make matching difficult. Coarse-to-fine methods use global context to guide local matching Yu et al. (2021); Qin et al. (2022); Li & Harada (2022a); Yu et al. (2023), often restricting fine correspondences to regions selected by coarse Top-K scores. This reduces computation but discards any correct pair outside the selected regions.

To preserve recoverable matches, we connect global correspondence search to high-resolution refinement through complete matching matrices. Diffusion establishes global hypotheses in a compact coarse space. Lifting transfers them to a structured fine-level source, and the conditional bridge iteratively updates the high-resolution matching matrix using fine-level geometry. The matching matrix itself evolves along this path: fine geometry can revise every candidate score, allowing refinement to correct coarse errors without excluding valid matches.

We propose BridgeMatch to realize this progressive expansion of matching-matrix space. Stage I estimates coarse correspondences by diffusion. Hierarchy-preserving lifting copies each score to its fine-level block, initializing Stage II without candidate pruning. The fine-stage predictor updates the full matching matrix and geometric alignment using point features and intermediate states. Figure 1 illustrates the formulation, and Fig. 2 shows the two-stage solver.

Within this unified formulation, we develop two complementary bridges. Endpoint-CFM learns a deterministic ODE along a paired straight path Lipman et al. (2023); Liu et al. (2022). The paired Brownian-bridge SDE combines endpoint and noise prediction to construct its forward drift De Bortoli et al. (2021); Tong et al. (2024). They share the lifted source, time-conditioned transformer, and endpoint-predictor architecture, while defining distinct transport dynamics. Each is trained with Stage I in a separate run.

Experiments demonstrate the effectiveness of this design in matching, registration, and transfer. The stochastic model reaches 82.41% NFMR and 78.79% IR on 4DLoMatch, exceeding the strongest

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.

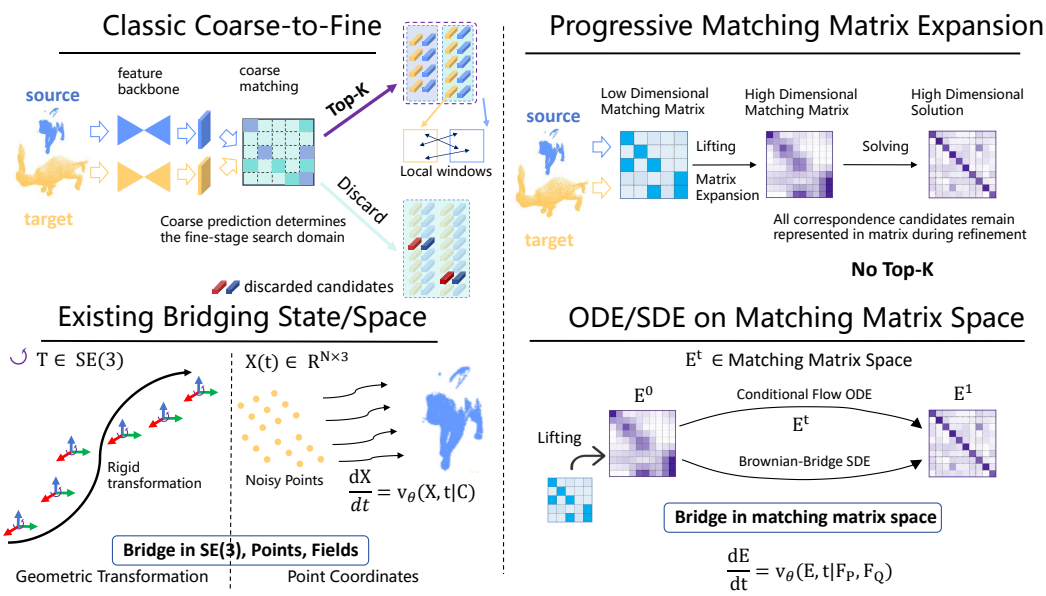


Figure 1: Comparison with existing paradigms. Traditional coarse-to-fine pipelines restrict fine matching through Top-K pruning, while our progressive expansion retains all candidates by lifting the complete coarse matrix. Unlike bridges in coordinate or transformation space, BridgeMatch transports the matching matrix ($\mathbf{E}_t$) through a conditional-flow ODE or a Brownian-bridge SDE.

listed baseline per metric by $5.26/5.49$ points. Both variants improve registration over Diff-Reg with GraphSCNet and transfer to CAPE and DeepDeform. Ablations further show that the model recovers correct correspondences excluded by coarse Top-K selection, improves matching through ODE iteration, and achieves effective refinement with a single SDE forward-drift update.

Our contributions are summarized as follows:

- To our knowledge, we introduce the first unified conditional transport framework in matching-matrix space for 3D deformable registration, with the matching matrix itself as the evolving state. A deterministic endpoint-CFM ODE and a stochastic paired Brownian-bridge SDE connect lifted coarse correspondences to high-resolution targets.
- We connect coarse diffusion for global correspondence search to high-resolution bridge refinement through hierarchy-preserving lifting. This progressive expansion transfers the complete coarse estimate as a structured source, retains all fine-level candidates, and allows geometry-guided updates to correct coarse matching errors.
- We demonstrate strong matching and registration performance on 4DMatch and 4DLoMatch and transfer to CAPE and DeepDeform. Controlled inference experiments establish the benefits of ODE iterative refinement and SDE forward-drift refinement, while candidate masking reveals the recovery advantage of retaining the full matching space.

2 Related Work

Point Cloud Registration. Recent registration methods improve feature learning, correspondence consistency, and deformation modeling. PARE-Net uses position-aware rotation-equivariant features to generate pose hypotheses from individual correspondences (Yao et al., 2024). CAST focuses cross-attention on candidate regions and uses geometric consistency in self-attention (Huang et al., 2024). For non-rigid alignment, NDP decomposes motion through an MLP pyramid with increasing positional frequencies (Li & Harada, 2022b); GraphSCNet rejects outliers using local spatial consistency on a deformation graph (Qin et al., 2023). SyNoRiM synchronizes learned functional maps to enforce cycle consistency across scans (Huang et al., 2022). OAR uses adaptive correntropy to reduce occlusion errors and local reconstruction to constrain unmatched regions (Zhao et al., 2025).

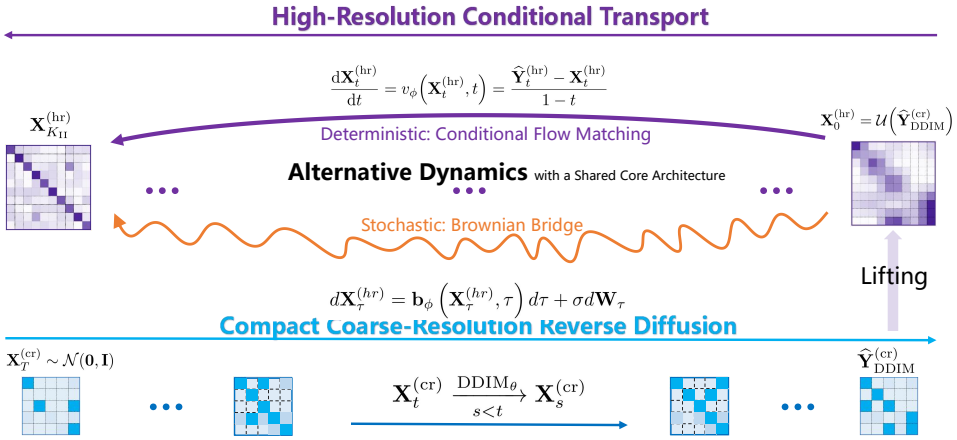


Figure 2: BridgeMatch architecture. Stage I estimates coarse matches by diffusion; lifting supplies the initial matrix for Stage II. Both Stage-II implementations use the same endpoint-predictor architecture; the stochastic model adds a noise head to construct the forward drift.

ERNet predicts deformation graph nodes and refines their trajectories in temporal windows (He et al., 2025). Chen et al. (2026) learn topology-aware propagation and codebook-enhanced superpoint features for deformation robustness. AniSym-Net couples anisotropic shape–motion fields with symplectic constraints to handle deformation and occlusion (Wang et al., 2025). RGGT combines pretrained generative features with correspondence–reconstruction supervision for rigid and non-rigid registration (Zheng et al., 2026). UniCorrn shares transformer weights across 2D/3D matching tasks, with separate appearance and position streams (Goswami et al., 2026).

Generative Registration. Generative registration refines matching matrices, poses, or coordinates. Diff-PCR searches doubly stochastic matrix space through iterative denoising (Shi & Wu, 2025). Diff-Reg supports 3D and 2D–3D matching with a lightweight denoiser that reuses backbone features across steps (Wu et al., 2024). ODIN alternates overlap discovery with correlation-matrix denoising (Jin et al., 2024). FUSER predicts multiview poses jointly; FUSER-DF refines them by diffusion in $\mathrm{SE}(3)^N$ (Jiang et al., 2026). Rectified Point Flow learns point-wise velocities for registration and shape assembly, then recovers poses from transported points (Sun et al., 2025). Register Any Point scales this coordinate-transport approach to multiview scenes with test-time rigidity enforcement (Pan et al., 2025). ODin generates aligned human point clouds with fixed point ordering to preserve template semantics (Masiero et al., 2026).

Diff-Reg refines a matching matrix by iterative denoising. BridgeMatch connects coarse diffusion to a structured fine-level source through hierarchical lifting, then learns geometry-conditioned ODE and SDE paths toward the target matrix. This design retains all candidate pairs and supports iterative correction at high resolution.

Flow Matching and Stochastic Bridges. Flow matching regresses conditional vector fields without simulating training trajectories (Lipman et al., 2023); rectified flow fits linear interpolations and uses reflow to straighten sampling paths (Liu et al., 2022). Riemannian flow matching constructs target fields from geometry-aware premetrics (Chen & Lipman, 2024). GaussianAnything separates geometry and appearance generation through cascaded flows in point-structured latent space (Lan et al., 2025). SGMatch combines semantic local attention with flow-based feature transport regularization for non-rigid correspondence (Ye et al., 2026). For stochastic transport, Tong et al. (2024) combine score and flow matching with entropy-regularized endpoint coupling to learn Schrödinger bridges without trajectory simulation. P2P-Bridge learns bridges from noisy to clean point coordinates (Vogel et al., 2024); BridgeShape connects incomplete and complete shapes in a depth-enhanced VQ-VAE latent space (Kong et al., 2026).

3 METHOD

3.1 METHOD OVERVIEW

Given source and target point clouds $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{N}$ and $\mathcal{Q} = \{\mathbf{q}_j\}_{j=1}^{M}$, we estimate a soft matching matrix for downstream non-rigid registration. A hierarchical backbone produces features $\mathbf{F}_s^{(r)}, \mathbf{F}_t^{(r)}$ at coarse ($r = \mathrm{cr}$) and high ($r = \mathrm{hr}$) resolutions. Binary matching matrices $\mathbf{Y}^{(r)} \in \{0,1\}^{N_r \times M_r}$ provide supervision, with positive entries marking ground-truth correspondences.

Stage I searches for global correspondences by denoising a compact matrix from Gaussian noise. Lifting transfers its scores to a structured source $\mathbf{X}_0$, from which Stage II uses fine geometry to refine all candidate scores (Fig. 2).

3.2 STAGE I: COARSE CORRESPONDENCE DIFFUSION

Forward diffusion adds Gaussian noise to the coarse target matrix. With cumulative signal coefficient $\bar{\alpha}_k$, the noisy matrix at step k is

$$\mathbf{X}_k^{(\mathrm{cr})} = \sqrt{\bar{\alpha}_k}\,\mathbf{Y}^{(\mathrm{cr})} + \sqrt{1-\bar{\alpha}_k}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{1}$$

A time-conditioned matching transformer uses this noisy matrix and coarse features to predict the clean endpoint:

$$\widehat{\mathbf{Y}}_{0,k}^{(\mathrm{cr})} = D_\theta\Big(\mathbf{X}_k^{(\mathrm{cr})}, k, \mathbf{F}_s^{(\mathrm{cr})}, \mathbf{F}_t^{(\mathrm{cr})}\Big). \tag{2}$$

A bounded confidence view of $\mathbf{X}_k^{(\mathrm{cr})}$ provides the geometric conditioning used by the transformer.

During inference, deterministic DDIM Song et al. (2020) starts from $\mathbf{X}_T^{(\mathrm{cr})} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and produces $\widehat{\mathbf{Y}}_{\mathrm{DDIM}}^{(\mathrm{cr})}$. Appendix B.1 gives the sampling updates.

3.3 LIFTING TO HIGH RESOLUTION

We detach the coarse prediction and transfer its entries using the point hierarchy. Let $\mathbf{P}_s \in \{0,1\}^{N_{\mathrm{hr}} \times N_{\mathrm{cr}}}$ and $\mathbf{P}_t \in \{0,1\}^{M_{\mathrm{hr}} \times M_{\mathrm{cr}}}$ be parent-assignment matrices, each with one nonzero entry per row. The lifted source is

$$\mathbf{X}_0 = \mathcal{U}(\widehat{\mathbf{Y}}^{(\mathrm{cr})}) = \mathbf{P}_s \widehat{\mathbf{Y}}^{(\mathrm{cr})} \mathbf{P}_t^\top. \tag{3}$$

Each coarse entry is copied to all high-resolution pairs sharing its parent points, retaining the complete candidate space. Fine-level geometric features resolve candidates within each block during refinement. Appendix B.2 gives the rank bound and a decomposition of the lifting error.

3.4 STAGE-II CONDITIONAL PATHS AND ENDPOINT PREDICTOR

Stage II learns from intermediate states between the source matrix and high-resolution target. During training, a curriculum selects $\mathbf{X}_0$ by varying coarse predictions and gradually removing ground-truth mixing. At inference, $\mathbf{X}_0$ is the lifted DDIM estimate. Conditioned on point geometry and backbone features, the training path is

$$\mathbf{X}_t^{(d)} = (1-t)\mathbf{X}_0 + t\mathbf{Y}^{(\mathrm{hr})} + \rho_d(t)\boldsymbol{\epsilon}, \quad \rho_d(t) = \begin{cases} 0, & d = \mathrm{ODE}, \\ \sigma_B\sqrt{t(1-t)}, & d = \mathrm{SDE}. \end{cases} \tag{4}$$

Here, $d \in \{\mathrm{ODE}, \mathrm{SDE}\}$ selects the path, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Below we omit hr: $\mathbf{X}_t$ denotes the current state and $\widehat{\mathbf{Y}}_t$ the predicted endpoint. In Fig. 2, $\mathbf{X}_{K_{\mathrm{II}}}^{(\mathrm{hr})}$ is the state after K_{II} solver steps.

The endpoint predictor first forms a bounded view $\mathbf{C}_t$ of the state. Endpoint-CFM uses its bounded state directly; the Brownian branch uses $\mathbf{C}_t = \mathrm{clip}(\mathbf{X}_t, 0, 1)$. After masking invalid pairs, Sinkhorn normalization gives assignment weights $\mathbf{A}_t$. Soft Procrustes then estimates

$$(\mathbf{R}_t, \boldsymbol{\tau}_t) = \arg\min_{\mathbf{R} \in \mathrm{SO}(3), \boldsymbol{\tau}} \sum_{ij} A_{t,ij} \|\mathbf{R}\mathbf{p}_i + \boldsymbol{\tau} - \mathbf{q}_j\|_2^2. \tag{5}$$

This transform supplies the geometric conditioning for the next prediction. The transformer processes the warped source $\widetilde{\mathbf{p}}_{i,t} = \mathbf{R}_t\mathbf{p}_i + \boldsymbol{\tau}_t$, target geometry, high-resolution features, and a sinusoidal time embedding:

$$\widehat{\mathbf{Y}}_t = H_\phi\Big(\mathbf{C}_t, t, \mathbf{F}_s^{(\mathrm{hr})}, \mathbf{F}_t^{(\mathrm{hr})}, \widetilde{\mathcal{P}}_t, \mathcal{Q}\Big). \tag{6}$$

The matrix $\mathbf{C}_t$ also biases the matching logits. Backbone features are computed once; the geometric alignment and endpoint prediction are updated at each step. In the stochastic branch, the noise head G_ψ and SDE update receive the unbounded $\mathbf{X}_t$, while clipping is confined to endpoint conditioning. Appendix B.3 gives the architecture.

3.5 Stage II-A: Deterministic Endpoint-CFM Solver

The deterministic model sets $\rho_{\mathrm{ODE}}(t) = 0$ in Eq. 4. The endpoint prediction defines the velocity

$$v_\phi(\mathbf{X}_t, t) = \frac{\widehat{\mathbf{Y}}_t - \mathbf{X}_t}{1-t}. \tag{7}$$

Positive correspondences occupy a small fraction of the matching matrix. We use focal endpoint supervision to emphasize misclassified entries and obtain the velocity from Eq. 7. On the straight training path, squared velocity error equals squared endpoint error weighted by $(1-t)^{-2}$.

Inference begins at the lifted DDIM matrix and solves

$$\frac{d\mathbf{X}_t}{dt} = v_\phi(\mathbf{X}_t, t), \qquad \mathbf{X}_0 = \mathcal{U}(\widehat{\mathbf{Y}}^{(\mathrm{cr})}_{\mathrm{DDIM}}). \tag{8}$$

We use explicit Euler steps with bounded updates and evaluate the field before the singular endpoint $t = 1$. The final Euler update equals the clipped endpoint prediction.

3.6 Stage II-B: Stochastic Paired Brownian-Bridge Solver

For the Brownian path, let $s_t = \sqrt{t(1-t)}$. The noise head G_ψ predicts $\widehat{\boldsymbol{\epsilon}}_t$ from the matching state, source matrix, time, and shared geometric features. Combined with the endpoint prediction, it gives the drift

$$b_\phi(\mathbf{X}_t, t) = \widehat{\mathbf{Y}}_t - \mathbf{X}_0 - \sigma_B\frac{t}{s_t}\widehat{\boldsymbol{\epsilon}}_t. \tag{9}$$

The sampler follows

$$\mathrm{d}\mathbf{X}_t = b_\phi(\mathbf{X}_t, t)\,\mathrm{d}t + \sigma_B\,\mathrm{d}\mathbf{W}_t, \tag{10}$$

where $\mathbf{W}_t$ is a standard matrix-valued Wiener process. At inference, we initialize $\mathbf{X}_\varepsilon = \mathbf{X}_0 + \sigma_B\sqrt{\varepsilon(1-\varepsilon)}\,\mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This matches the training path's noise variance at ε and centers the state at the available source $\mathbf{X}_0$. Euler–Maruyama steps over $[\varepsilon, 1-\varepsilon]$ are followed by a final endpoint prediction. With exact endpoint and noise estimates on the paired training path, the drift reduces to $(\mathbf{Y} - \mathbf{X}_t)/(1-t)$.

Each run jointly trains Stage I and one Stage-II branch with focal correspondence and geometric motion losses. The stochastic branch averages noise errors separately over positive and negative pairs, giving each class equal total weight. Endpoint and noise predictions define the drift through Eq. 9. Coarse predictions are detached before lifting; both correspondence losses update the shared backbone. Appendix B.4 details training.

4 Experiments

4.1 Protocol

Datasets. We evaluate matching on 4DMatch and 4DLoMatch Li & Harada (2022a), which contain partial point-cloud pairs with non-rigid motion. A 45% overlap boundary separates the test splits, with 4DLoMatch covering lower overlap. Downstream registration uses the filtered 4DMatch-F/4DLoMatch-F splits Li & Harada (2022b); Qin et al. (2023).

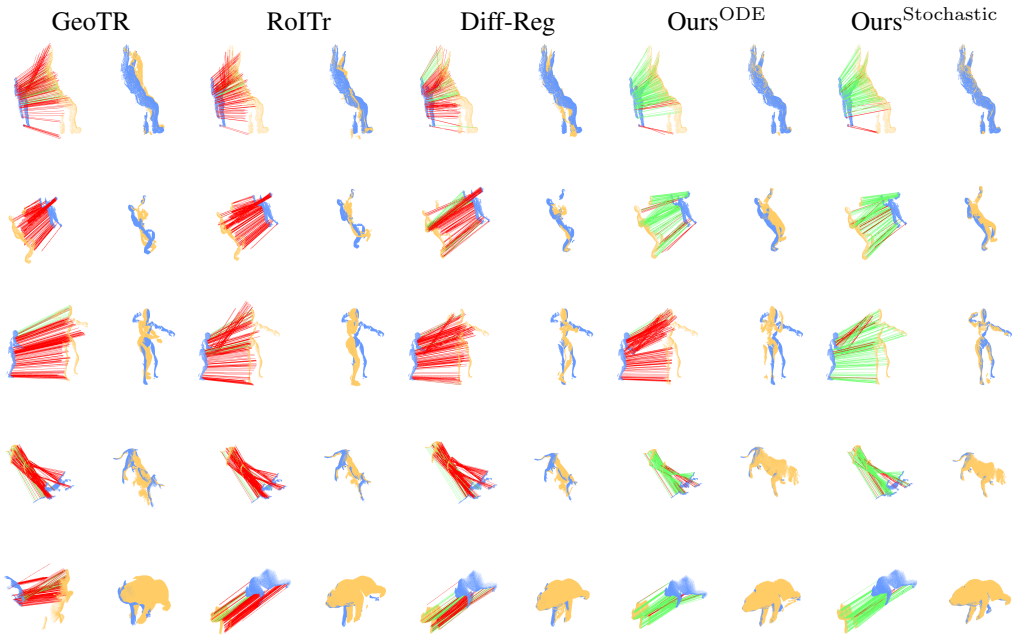


Figure 3: Qualitative non-rigid registration comparisons on 4DMatch/4DLoMatch. For each method, the left image visualizes predicted correspondences and the right image shows the deformation estimated by GraphSCNet. Blue and yellow denote the source and target point clouds, respectively; green and red lines indicate correct and incorrect correspondences. ODE and Stochastic denote our deterministic endpoint-CFM and stochastic Brownian-bridge variants, respectively. Zoom in for details.

Metrics. We report non-rigid feature matching recall (NFMR) and inlier ratio (IR) at a $4\,\mathrm{cm}$ threshold. NFMR measures ground-truth match recovery through interpolation from predicted matches. IR is the fraction of predicted matches correct under the ground-truth warp. For registration, endpoint error (EPE) averages distances between predicted and ground-truth warped points. Strict and relaxed accuracy (AccS/AccR) measure the fractions of points meeting their absolute-or-relative error thresholds. Outlier ratio (OR) measures the fraction of points with relative error above 30%. EPE is in meters; other metrics are percentages. Lower EPE/OR and higher accuracy or recall indicate better performance.

Zero-shot protocol. Directed test pairs come from MPC-CAPE and MPC-DD, the preprocessed CAPE and DeepDeform datasets released with SyNoRiM Huang et al. (2022). Diff-Reg and our models use 4DMatch-only checkpoints and the same fixed 4DMatch-trained GraphSCNet on both datasets, without target-domain adaptation. The matched DeepDeform evaluation also includes Lepard. Stochastic results are averaged over sampling seeds $\{0, 1, 2\}$; deterministic matchers run once unless noted. The symbols * and † mark target-domain training and different evaluation protocols, respectively. AccS/AccR thresholds are $(2\,\mathrm{cm}, 5\%)/(5\,\mathrm{cm}, 10\%)$ on CAPE and $(2.5\,\mathrm{cm}, 2.5\%)/(5\,\mathrm{cm}, 5\%)$ on DeepDeform. Metrics are averaged over test pairs.

Training and inference settings. We use Lepard's KPFCN hierarchy and train for 60 epochs with batch size 2 and SGD (learning rate 0.015, momentum 0.93, weight decay 10^{-6}, exponential decay 0.95, gradient clipping at 5). Both solvers default to 5 coarse DDIM and 20 fine bridge steps; SDE uses $\sigma_B = 0.1$ and $\varepsilon = 0.02$. We extract correspondences by mutual nearest matching with confidence threshold 0.2. The ODE and SDE models are trained separately with matched epochs and step counts but different paths, auxiliary heads, and losses.

Table 1: Non-rigid correspondence on 4DMatch/4DLoMatch (%). Bold/underline mark best/second-best. SDE drift ODE: lifted source, one update, 2 Stage-II NFE; initial noise retained, subsequent increments omitted.

	4DMatch		4DLoMatch	
Method	NFMR ↑	IR ↑	NFMR ↑	IR ↑
PointPWC Wu et al. (2019)	21.60	20.00	10.00	7.20
FLOT Puy et al. (2020)	27.10	24.90	15.20	10.70
D3Feat Bai et al. (2020)	55.50	54.70	27.40	21.50
Predator Huang et al. (2021)	56.40	60.40	32.10	27.50
Lepard Li & Harada (2022a)	83.60	82.64	66.63	55.55
GeoTransformer Qin et al. (2022)	83.20	82.20	65.40	63.60
RoITr Yu et al. (2023)	83.00	84.40	69.40	67.60
Diff-Reg Wu et al. (2024)	90.25	87.98	77.15	67.00
RoITr-WN Yanagi et al. (2024)	87.2	87.3	75.3	73.3
Ours^{ODE}	91.47	89.82	78.64	71.78
Ours^{Stochastic}	<u>92.43</u>	<u>91.47</u>	<u>82.41</u>	<u>78.79</u>
Ours$^{\text{SDE drift ODE}}$	**93.10**	**92.25**	**83.94**	**80.49**

Table 2: Non-rigid registration on 4DMatch-F/4DLoMatch-F. EPE is in meters; AccS/AccR/OR are percentages. GraphSCNet-labeled entries use this deformation solver Qin et al. (2023); others retain their respective pipelines. RGGT's TF denotes pretrained TRELLIS features; the gray row uses this additional prior and is excluded from ranking. Bold/underline mark best/second-best among the remaining rows.

Method	4DMatch-F				4DLoMatch-F			
	EPE (m)↓	AccS↑	AccR↑	OR↓	EPE (m)↓	AccS↑	AccR↑	OR↓
NSEFP Li et al. (2021)	0.265	8.7	18.7	65.0	0.495	0.4	1.6	84.8
Nerfies Park et al. (2021)	0.280	12.7	25.4	58.9	0.498	1.1	3.0	82.2
PointPWC-Net Wu et al. (2020)	0.182	6.3	21.5	52.1	0.279	1.7	8.2	55.7
FLOT Puy et al. (2020)	0.133	7.7	27.2	40.5	0.210	2.7	13.1	42.5
DGFM Donati et al. (2020)	0.152	12.3	32.6	37.9	0.148	1.9	6.5	64.6
SyNoRiM Huang et al. (2022)	0.099	22.9	49.9	26.0	0.170	10.6	30.2	31.1
NDP Li & Harada (2022b)	0.077	61.3	74.1	17.3	0.177	26.6	41.1	33.8
RoITrYu et al. (2023) +GraphSCNet	0.056	59.60	80.50	12.50	0.118	32.30	56.70	20.50
LepardLi & Harada (2022a)+GraphSCNet	0.042	70.10	83.80	9.20	0.102	40.00	59.10	17.50
GeoTRQin et al. (2022)+GraphSCNet	0.043	72.10	84.30	9.50	0.119	41.00	58.40	20.60
Diff-RegWu et al. (2024)+GraphSCNet	0.041	<u>73.20</u>	85.80	8.30	0.095	43.80	62.90	15.50
Lepard+OARZhao et al. (2025)	0.059	59.32	74.33	16.41	0.251	27.25	42.01	45.04
RGGTZheng et al. (2026) (w/o TF)	0.039	62.10	79.60	11.3	–	–	–	–
RGGTZheng et al. (2026) (w TF)	0.031	73.9	89.7	7.1	0.052	55.0	79.0	6.8
Ours^{ODE}+GraphSCNet	<u>0.037</u>	73.1	<u>86.3</u>	<u>7.9</u>	<u>0.094</u>	<u>45.0</u>	<u>65.0</u>	<u>14.9</u>
Ours^{Stochastic}+GraphSCNet	**0.036**	**74.1**	**87.0**	**7.4**	**0.089**	**47.2**	**67.2**	**13.8**

4.2 Non-Rigid Correspondence and Registration

Both bridge designs improve on Diff-Reg in all four matching metrics (Table 1). A single SDE forward-drift update reaches $93.10/92.25$ NFMR/IR on 4DMatch and $83.94/80.49$ on 4DLoMatch with 2 Stage-II NFE, leading all four columns. The forward drift combines endpoint guidance with learned noise correction to refine the matching state toward the target. These results demonstrate the SDE bridge's effectiveness for accurate few-step refinement on 4DMatch and 4DLoMatch.

Improved correspondences translate into better registration with GraphSCNet (Table 2). Relative to Diff-Reg, SDE lowers EPE from 0.041 to 0.036 on 4DMatch-F and from 0.095 to 0.089 on 4DLoMatch-F, with higher AccS/AccR and lower OR on both. ODE also lowers EPE on both splits. Figure 3 illustrates matching and alignment around limbs and extremities.

Table 3: CAPE generalization (EPE in m; other metrics in %). Diff-Reg and ours are evaluated on 2,508 pairs. *: target-domain training; –: unverified information. Bold/underline mark best/second-best only in the No group. Stochastic results average three seeds; training and input settings differ across methods.

Method	Venue	Training data	Target train	EPE (m)↓	AccS↑	AccR↑	OR↓
Target-domain training							
SyNoRiM (self-sup., four-view)* Huang et al. (2022)	TPAMI 2022	CAPE	Yes	0.030	55.5	89.1	59.1
Target-domain training status unverified							
PointPWC-Net Wu et al. (2020)	ECCV 2020	–	–	0.039	17.9	35.7	85.9
FLOT Puy et al. (2020)	ECCV 2020	–	–	0.049	21.2	31.1	92.2
DGFM Donati et al. (2020)	CVPR 2020	–	–	0.036	35.5	62.4	73.7
Lepard+N-ICP Li & Harada (2022a)	CVPR 2022	–	–	0.089	23.9	44.7	78.0
LNDP Li & Harada (2022b)	NeurIPS 2022	–	–	0.045	61.3	92.2	45.7
Zero-shot transfer (no target-domain training)							
GeoTransformer Qin et al. (2022)	CVPR 2022	ModelNet	No	0.055	20.9	50.1	75.5
GraphSCNet Qin et al. (2023)	CVPR 2023	4DMatch	No	0.059	66.7	79.4	62.1
Diff-Reg Wu et al. (2024) + GraphSCNet	ECCV 2024	4DMatch	No	0.018	78.4	95.1	45.4
RGGT (w TF) Zheng et al. (2026)	ICML 2026	ModelNet + 4DMatch	No	0.023	77.7	93.2	**36.4**
Ours^{ODE}+ GraphSCNet		4DMatch	No	<u>0.017</u>	<u>79.2</u>	<u>95.3</u>	44.7
Ours^{Stochastic}+ GraphSCNet		4DMatch	No	**0.016**	**80.4**	**95.7**	<u>43.4</u>

Table 4: DeepDeform generalization (EPE in m; other metrics in %). † marks cited results under different protocols; all other rows are evaluated on 1,221 pairs after 1 cm near-rigid filtering. * denotes target-domain training or fine-tuning. Bold/underline mark best/second-best only in the No group. Stochastic results average three seeds.

Method	Venue	Training data	Target train	EPE (m)↓	AccS↑	AccR↑	OR↓
Target-domain training							
SyNoRiM* Huang et al. (2022)	TPAMI 2022	MPC-DD	Yes	0.0268	72.9	91.8	12.8
AniSym-Net*† Wang et al. (2025)	TPAMI 2025	MPI-FAUST → DeepDeform	Yes	0.0266	–	89.4	–
Zero-shot transfer (no target-domain training)							
GeoTransformer Qin et al. (2022)+GraphSCNet†	CVPR 2022	4DMatch	No	0.134	24.1	44.2	59.1
Lepard Li & Harada (2022a) + GraphSCNet	CVPR 2022	4DMatch	No	0.1501	17.1	35.1	64.0
Diff-Reg Wu et al. (2024) + GraphSCNet	ECCV 2024	4DMatch	No	0.0748	43.6	66.1	36.0
Ours^{ODE} + GraphSCNet		4DMatch	No	<u>0.0697</u>	<u>46.4</u>	<u>69.4</u>	<u>33.8</u>
Ours^{Stochastic} + GraphSCNet		4DMatch	No	**0.0620**	**53.2**	**74.9**	**28.3**

4.3 Zero-Shot Generalization on CAPE

Following RGGT Zheng et al. (2026), we expand 209 four-frame MPC-CAPE samples into 12 ordered pairs each, giving 2,508 test pairs.

Both bridge designs improve CAPE transfer over Diff-Reg using the same GraphSCNet solver (Table 3). ODE and SDE reduce EPE from 0.018 to 0.017 and 0.016, respectively. SDE also raises AccS/AccR to $80.4/95.7\%$. Across all three deformation groups, SDE achieves the best EPE, AccS, and OR among the evaluated methods (supplementary material), showing consistent transfer across motion magnitudes.

SyNoRiM uses target-domain self-supervision and four-view input, and RGGT adds training data and TRELLIS features. Our SDE model leads the zero-shot group in EPE, AccS, and AccR; RGGT has the lowest OR.

4.4 Zero-Shot Generalization on DeepDeform

We evaluate 1,221 directed MPC-DD/DeepDeform pairs after near-rigid filtering. We fit a rigid transform on flow-visible points and exclude pairs with mean residual below 1 cm over the full source cloud. The published GraphSCNet split contains 1,011 pairs. Reference rows are grouped by training and evaluation protocol.

SDE achieves the best results among the matched zero-shot methods on all four metrics (Table 4). Compared with Diff-Reg, EPE falls from 0.0748 to 0.0620 (17.1%), AccS/AccR rise by $9.6/8.8$ points, and OR falls by 7.7 points. ODE also improves all four metrics, demonstrating that both bridge designs transfer effectively without target-domain adaptation.

Table 5: Training and inference-source controls for the stochastic model on 4DLoMatch (%). Results average three sampling seeds per trained model. The last two rows use the same full-model checkpoint; high-resolution inference uses 20 updates. Coarse-only predictions are lifted for evaluation.

Training	Inference source / output	NFMR↑	IR↑
Coarse-only	Lifted coarse output	34.04	18.03
High-res.-only	$\mathcal{U}(0,1)$ source	80.27	75.98
Full two-stage	Lifted Stage-I source	82.41	78.79
Full two-stage (same weights)	$\mathcal{U}(0,1)$ source	82.43	78.92

Table 6: Inference controls on one SDE-trained checkpoint, 4DLoMatch (%). Within each source, all modes share the initial perturbation. Forward-drift and probability-flow updates omit subsequent random increments and include a final endpoint prediction. Bold marks the higher scores at 2 NFE within each source.

Source	Inference mode	Stage-II NFE	NFMR↑	IR↑
Lifted	Direct endpoint	1	79.28	76.69
Lifted	Forward drift	2	**83.94**	**80.49**
Lifted	Probability flow	2	74.06	72.56
Uniform	Direct endpoint	1	75.12	72.79
Uniform	Forward drift	2	**82.82**	**79.61**
Uniform	Probability flow	2	74.85	73.02

4.5 Ablations and Solver Budgets

Multiresolution design and initialization. The full SDE design gains $2.14/2.81$ NFMR/IR points over high-resolution-only (Table 5). At twenty updates, lifted and uniform sources reach $82.41/78.79$ and $82.43/78.92$. Lifting benefits one-update ODE/SDE inference; longer refinement performs well from either source (Appendix C.2).

SDE iterative refinement. The SDE bridge delivers effective few-step refinement through its learned forward drift (Table 6). One drift update followed by endpoint prediction raises lifted-source NFMR/IR from $79.28/76.69$ to $83.94/80.49$, using 2 versus 1 NFE. At the same 2 NFE, forward drift exceeds probability flow by $9.88/7.93$ points. This advantage holds at every tested budget (Appendix C.9), highlighting the SDE forward drift's strength in few-step correspondence refinement.

ODE and SDE refinement budgets. ODE refinement raises NFMR/IR from $75.76/70.11$ at one step to $78.46/71.71$ at ten. SDE sampling peaks at two updates with $83.16/79.62$. These budgets take 355 and 64 ms in Stage II, versus 740 and 639 ms at twenty updates, demonstrating efficient refinement (Appendix C.8).

Candidate coverage. Full-candidate refinement recovers correct matches excluded by coarse selection. Top-256 retains 34.23% of supervised pairs; unrestricted inference recovers 49.30% of the excluded pairs and gains 33.10 NFMR points (Appendix C.3). This fixed-model masking experiment uses dense computation; the matching state requires $O(N_{\mathrm{hr}} M_{\mathrm{hr}})$ memory.

5 Conclusion

BridgeMatch unifies ODE and SDE transport with the matching matrix as the evolving state. Hierarchical lifting connects coarse global search to full-candidate refinement, with endpoint and class-balanced noise supervision for sparse matching targets. Experiments demonstrate matching and transfer gains, recovery of correct matches excluded by coarse Top-K selection, effective ODE iteration, and accurate single-update SDE drift refinement. Measured costs establish the efficiency of reduced-step inference.

AI USE STATEMENT

We used generative AI tools for language editing, LaTeX formatting, literature searches, and manuscript organization. The authors developed the methodology, implemented the model, conducted the experiments, and analyzed the results. The authors reviewed the AI-assisted text and take responsibility for the paper and accompanying materials.

ETHICS STATEMENT

This work uses public datasets under their respective licenses and does not involve new human-subject data collection. The method is intended for vision and robotics research and is subject to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

The paper and supplement describe the architecture, training losses, sampling updates, and evaluation protocols. All datasets are public. Upon acceptance, we will release the implementation, trained weights, and data preparation instructions.

REFERENCES

Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Haozhe Chen, Rui Li, Zhengbao Wang, Xinhao Zhu, Linjie Li, Tianyu Xiong, Xuan Ouyang, and Jiaqi Yang. Topology-aware feature propagation for unsupervised non-rigid point cloud correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31389–31399, 2026.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, 2024.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8589–8598. IEEE, 2020.

Prajnan Goswami, Tianye Ding, Feng Liu, and Huaizu Jiang. Unicorrn: Unified correspondence transformer across 2d and 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9943–9954, June 2026.

Guangzhao He, Yuxi Xiao, Zhen Xu, Xiaowei Zhou, and Sida Peng. ERNet: Efficient non-rigid registration network for point sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27156–27165, 2025.

Jiahui Huang, Tolga Birdal, Zan Gojcic, Leonidas J Guibas, and Shi-Min Hu. Multiway non-rigid point cloud registration via learned functional map synchronization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2038–2053, 2022.

Renlang Huang, Yufan Tang, Jiming Chen, and Liang Li. A consistency-aware spot-guided transformer for versatile and hierarchical point cloud registration. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2021.

Haobo Jiang, Jin Xie, Jian Yang, Liang Yu, and Jianmin Zheng. FUSER: Feed-forward multiview 3D registration transformer and $\mathrm{SE}(3)^N$ diffusion refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7393–7403, 2026.

Shengze Jin, Iro Armeni, Marc Pollefeys, and Daniel Barath. Multiway point cloud mosaicking with diffusion and global optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20838–20849, 2024.

Dequan Kong, Honghua Chen, Zhe Zhu, and Mingqiang Wei. BridgeShape: Latent diffusion schr"odinger bridge for 3D shape completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(7):5726–5734, 2026. doi: 10.1609/aaai.v40i7.37493.

Yushi Lan, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy. GaussianAnything: Interactive point cloud flow matching for 3D generation. In *International Conference on Learning Representations*, 2025.

Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems*, 34(4):1533–1574, 2014. doi: 10.3934/dcds.2014.34.1533.

Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *Advances in Neural Information Processing Systems*, 2021.

Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022a.

Yang Li and Tatsuya Harada. Non-rigid point cloud registration with neural deformation pyramid. *Advances in Neural Information Processing Systems*, 2022b.

Xingyu Lin, Carl Qi, Yunchu Zhang, Zhiao Huang, Katerina Fragkiadaki, Yunzhu Li, Chuang Gan, and David Held. Planning with spatial-temporal abstraction from point clouds for deformable object manipulation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1640–1651, 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. URL `https://openreview.net/forum?id=PqvMRDCJT9t`.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Mattia Masiero, Ilya A. Petrov, Daniel Cremers, Gerard Pons-Moll, and Riccardo Marin. Ordered diffusion for 3D human registration. *arXiv preprint arXiv:2608.05804*, 2026.

Yue Pan, Tao Sun, Liyuan Zhu, Lucas Nunes, Iro Armeni, Jens Behley, and Cyrill Stachniss. Register any point: Scaling 3D point cloud registration by flow matching. *arXiv preprint arXiv:2512.01850*, 2025.

Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5845–5854. IEEE, 2021.

Sergey Prokudin, Qianli Ma, Maxime Raafat, Julien Valentin, and Siyu Tang. Dynamic point fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7964–7976, 2023.

Gilles Puy, Alexandre Boulch, and Renaud Marlet. Flot: Scene flow on point clouds guided by optimal transport. In *European conference on computer vision*, pp. 527–544. Springer, 2020.

Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Zheng Qin, Hao Yu, Changjian Wang, Yuxing Peng, and Kai Xu. Deep graph-based spatial consistency for robust non-rigid point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Haihua Shi and Qianliang Wu. Diff-pcr: Diffusion-based correspondence searching in doubly stochastic matrix space for point cloud registration, 2025.

Haochen Shi, Huazhe Xu, Zhiao Huang, Yunzhu Li, and Jiajun Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks. In *Proceedings of Robotics: Science and Systems*, 2022. doi: 10.15607/RSS.2022.XVIII.008.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Tao Sun, Liyuan Zhu, Shengyu Huang, Shuran Song, and Iro Armeni. Rectified point flow: Generic point cloud pose estimation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL `https://rectified-pointflow.github.io/`.

Alexander Y. Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 1279–1287, 2024.

Mathias Vogel, Keisuke Tateno, Marc Pollefeys, Federico Tombari, Marie-Julie Rakotosaona, and Francis Engelmann. P2P-Bridge: Diffusion bridges for 3D point cloud denoising. In *European Conference on Computer Vision*, 2024.

Jinyang Wang, Xuequan Lu, Mohammed Bennamoun, and Bin Sheng. Non-rigid point cloud registration via anisotropic hybrid field harmonization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(9):7898–7915, 2025. doi: 10.1109/TPAMI.2025.3572584.

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*, pp. 160–178. Springer, 2024.

Wenxuan Wu, Zhiyuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *arXiv preprint arXiv:1911.12408*, 2019.

Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *European conference on computer vision*, pp. 88–107. Springer, 2020.

Rintaro Yanagi, Atsushi Hashimoto, Naoya Chiba, Shusaku Sone, Jiaxin Ma, and Yoshitaka Ushiku. Learning 3d point cloud registration as a single optimization problem. In *Asian Conference on Computer Vision*, pp. 175–193. Springer, 2024.

Zixin Yang, Richard Simon, Kelly Merrell, and Cristian A. Linte. Boundary constraint-free biomechanical model-based surface matching for intraoperative liver deformation correction. *IEEE Transactions on Medical Imaging*, 44(4):1723–1734, 2025. doi: 10.1109/TMI.2024.3515632.

Runzhao Yao, Shaoyi Du, Wenting Cui, Canhui Tang, and Chengwu Yang. PARE-Net: Position-aware rotation-equivariant networks for robust point cloud registration. In *European Conference on Computer Vision*, pp. 287–303. Springer, 2024.

Tianwei Ye, Xiaoguang Mei, Yifan Xia, Fan Fan, Jun Huang, and Jiayi Ma. SGMatch: Semantic-guided non-rigid shape matching with flow regularization. *arXiv preprint arXiv:2603.12937*, 2026.

Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *Advances in Neural Information Processing Systems*, 2021.

Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Mingyang Zhao, Gaofeng Meng, and Dong-ming Yan. Occlusion-aware non-rigid point cloud registration via unsupervised neural deformation correntropy. In *International Conference on Learning Representations*, volume 2025, pp. 33351–33375, 2025.

Chengyu Zheng, Songlin Yang, Jin Huang, Honghua Chen, Weiming Wang, Haoran Xie, Fu Lee Wang, and Mingqiang Wei. Rggt: A generative-prior-guided transformer for unified rigid and non-rigid point cloud registration. In *Forty-third International Conference on Machine Learning*, 2026.

Jing Zou, Bingchen Gao, Youyi Song, and Jing Qin. A review of deep learning-based deformable medical image registration. *Frontiers in Oncology*, 12:1047215, 2022. doi: 10.3389/fonc.2022.1047215.

SUPPLEMENTARY MATERIAL

A TECHNICAL BACKGROUND

We summarize the path definitions used by our solvers. Let $z = (\mathbf{X}_0, \mathbf{Y}) \sim \pi_c$ denote paired matrices under geometric condition c, and let $t \in [0, 1]$.

A.1 FLOW MATCHING

Conditional Flow Matching learns a velocity from an analytic conditional path without simulating training trajectories Lipman et al. (2023). For conditional velocity u_t, its squared objective is

$$\mathcal{L}_{\mathrm{CFM}} = \mathbb{E}_{t,z,\mathbf{X}_t}\left[\|v_\phi(\mathbf{X}_t, t, c) - u_t(\mathbf{X}_t \mid z, c)\|_F^2\right]. \tag{11}$$

The population optimum of the squared loss is $v_t^\star(\mathbf{X}, c) = \mathbb{E}[u_t \mid \mathbf{X}_t = \mathbf{X}, c]$. For the straight path $\mathbf{X}_t = (1-t)\mathbf{X}_0 + t\mathbf{Y}$, $u_t = \mathbf{Y} - \mathbf{X}_0$ Liu et al. (2022). Our endpoint-CFM model defines the velocity as $v_\phi = (\widehat{\mathbf{Y}}_t - \mathbf{X}_t)/(1-t)$ and trains the endpoint predictor with focal correspondence loss. Section B.2 relates squared velocity error to endpoint error. Section B.4 gives the training loss.

A.2 SCHRÖDINGER BRIDGES

Given reference diffusion $\mathbb{P}^\sigma$ and endpoint marginals μ_0, μ_1, the classical Schrödinger bridge minimizes the Kullback–Leibler divergence between path distributions Léonard (2014); De Bortoli et al. (2021):

$$\mathbb{Q}^\star = \arg\min_{\mathbb{Q}} D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}^\sigma), \qquad \mathbb{Q}_0 = \mu_0, \quad \mathbb{Q}_1 = \mu_1. \tag{12}$$

Our paired Brownian construction fixes the coupling through supervised source–target pairs. Conditioned on an endpoint pair $z = (\mathbf{X}_0, \mathbf{Y})$, the Brownian path has noise scale $\rho(t) = \sigma_B s_t$, where $s_t = \sqrt{t(1-t)}$ Tong et al. (2024). Writing q_t for this endpoint-conditioned density, its score and drift satisfy

$$\nabla_{\mathbf{X}_t} \log q_t = -\frac{\boldsymbol{\epsilon}}{\sigma_B s_t}, \qquad b_t = u_t + \frac{\sigma_B^2}{2}\nabla_{\mathbf{X}_t} \log q_t. \tag{13}$$

These identities hold for the given endpoint pair and guide the design of the Brownian sampler. We train the endpoint and noise predictors with the losses in Sec. B.4.

A.3 A SHARED PATH FAMILY

Both branches use the following path between the source matrix $\mathbf{X}_0$ and target matrix $\mathbf{Y}$:

$$\mathbf{X}_t = (1-t)\mathbf{X}_0 + t\mathbf{Y} + \rho(t)\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{14}$$

Holding the endpoints and noise fixed gives $u_t = \mathbf{Y} - \mathbf{X}_0 + \dot{\rho}(t)\boldsymbol{\epsilon}$. Zero variance gives endpoint-CFM; $\rho(t) = \sigma_B\sqrt{t(1-t)}$ gives the Brownian path. With exact fields, suitable regularity, and the same initial marginal at an interior time, the Brownian SDE and its probability-flow ODE share time marginals. The zero-variance branch follows a different path. Section B.4 describes how the task losses train the two predictors.

B SOLVER AND TRAINING DETAILS

Model and notation. BridgeMatch estimates correspondences between source points $\{\mathbf{p}_i\}$ and target points $\{\mathbf{q}_j\}$. A shared backbone produces coarse (cr) and high-resolution (hr) features. Stage I denoises a coarse matching matrix; Stage II refines a high-resolution matrix. The binary matrices $\mathbf{Y}^{(r)}$ provide correspondence supervision at resolution r. One-hot parent assignments $\mathbf{P}_s, \mathbf{P}_t$ lift coarse scores to

$$\mathbf{X}_0 = \mathcal{U}(\widehat{\mathbf{Y}}^{(\mathrm{cr})}) = \mathbf{P}_s \widehat{\mathbf{Y}}^{(\mathrm{cr})} \mathbf{P}_t^\top. \tag{15}$$

Each coarse entry is copied to pairs with the same parents, keeping all candidates. We omit hr when describing Stage II. During training, $\mathbf{X}_{0,\mathrm{tr}}$ denotes the source selected by the training schedule in

Sec. B.4. In the training-path formulas, $\mathbf{X}_0$ is shorthand for this source. At inference, it denotes the lifted DDIM estimate unless the experiment replaces the source. The two bridge variants have separate weights. Both use H_ϕ to predict the endpoint, and the stochastic model also uses G_ψ to predict noise.

B.1 Solver Parameterization and Sampling Details

Stage-I DDIM sampling. Let $\bar{\alpha}_k = \prod_{j=1}^{k}(1-\beta_j)$ be the cumulative signal coefficient, where β_j is the forward diffusion noise variance at step j. The coarse training state is $\mathbf{X}_k^{(\mathrm{cr})} = \sqrt{\bar{\alpha}_k}\mathbf{Y}^{(\mathrm{cr})} + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}$, with standard Gaussian noise $\boldsymbol{\epsilon}$. A time-conditioned matching transformer predicts the clean matrix $\widehat{\mathbf{Y}}_{0,k}^{(\mathrm{cr})}$ from this state and coarse point features. At inference, $\mathbf{X}_T^{(\mathrm{cr})} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is updated with deterministic DDIM Song et al. (2020). From the clean prediction, the estimated noise is

$$\widehat{\boldsymbol{\epsilon}}_k = \frac{\mathbf{X}_k^{(\mathrm{cr})} - \sqrt{\bar{\alpha}_k}\widehat{\mathbf{Y}}_{0,k}^{(\mathrm{cr})}}{\sqrt{1-\bar{\alpha}_k}}, \tag{16}$$

and the $\eta = 0$ update to $k' < k$ is

$$\mathbf{X}_{k'}^{(\mathrm{cr})} = \sqrt{\bar{\alpha}_{k'}}\widehat{\mathbf{Y}}_{0,k}^{(\mathrm{cr})} + \sqrt{1-\bar{\alpha}_{k'}}\widehat{\boldsymbol{\epsilon}}_k. \tag{17}$$

The final prediction $\widehat{\mathbf{Y}}_{\mathrm{DDIM}}^{(\mathrm{cr})}$ is lifted to form the Stage-II source.

Deterministic path and Euler integration. The deterministic model sets $\rho_{\mathrm{ODE}}(t) = 0$ in Eq. 14, giving the straight conditional path

$$\mathbf{X}_t = (1-t)\mathbf{X}_0 + t\mathbf{Y}^{(\mathrm{hr})}, \tag{18}$$

We use $K_{\mathrm{II}} = 20$ explicit Euler steps. With $t_k = k/K_{\mathrm{II}}$ and $\Delta t = 1/K_{\mathrm{II}}$, we evaluate

$$\widehat{\mathbf{Y}}_k = H_\phi(\mathbf{X}_k, t_k, \text{geometry/features}), \quad \mathbf{X}_{k+1} = \mathrm{clip}\left(\mathbf{X}_k + \Delta t\,\frac{\widehat{\mathbf{Y}}_k - \mathbf{X}_k}{1-t_k}, 0, 1\right). \tag{19}$$

We evaluate the field at $t_0, \ldots, t_{K_{\mathrm{II}}-1}$, before the singular endpoint. The last update equals the clipped endpoint prediction (Sec. B.2). Earlier updates change the matrix used for geometric conditioning.

Brownian path, noise head, and drift construction. The stochastic model sets $\rho_{\mathrm{SDE}}(t) = \sigma_B\sqrt{t(1-t)}$, giving the paired Brownian-bridge path

$$\mathbf{X}_t = (1-t)\mathbf{X}_0 + t\mathbf{Y}^{(\mathrm{hr})} + \sigma_B s_t \boldsymbol{\epsilon}, \quad s_t = \sqrt{t(1-t)}, \tag{20}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. A noise head predicts the injected unit noise from the current state, lifted source, time, and shared transformer features:

$$\widehat{\boldsymbol{\epsilon}}_t = G_\psi(\mathbf{X}_t, \mathbf{X}_0, t, \mathbf{f}_s, \mathbf{f}_t). \tag{21}$$

Here, $\mathbf{f}_s$ and $\mathbf{f}_t$ are the source and target features shared with the endpoint head. The noise head uses the unclipped matching states. Section B.3 summarizes its design.

Endpoint and noise estimates define a conditional velocity and scaled score,

$$v_\phi(\mathbf{X}_t, t) = \widehat{\mathbf{Y}}_t - \mathbf{X}_0 + \sigma_B\frac{1-2t}{2s_t}\widehat{\boldsymbol{\epsilon}}_t, \quad \widetilde{s}_\phi(\mathbf{X}_t, t) = -\widehat{\boldsymbol{\epsilon}}_t/s_t. \tag{22}$$

The scaled score and velocity give

$$b_\phi = v_\phi + \frac{\sigma_B}{2}\widetilde{s}_\phi. \tag{23}$$

Substitution gives the sampling dynamics

$$b_\phi(\mathbf{X}_t, t) = \widehat{\mathbf{Y}}_t - \mathbf{X}_0 - \sigma_B\frac{t}{s_t}\widehat{\boldsymbol{\epsilon}}_t, \tag{24}$$

$$\mathrm{d}\mathbf{X}_t = b_\phi(\mathbf{X}_t, t)\,\mathrm{d}t + \sigma_B\,\mathrm{d}\mathbf{W}_t, \tag{25}$$

where $\mathbf{W}_t$ is a standard matrix-valued Wiener process.

Stochastic initialization and integration. For sampling, we use $\mathbf{X}_0 = \mathcal{U}(\widehat{\mathbf{Y}}^{(\mathrm{cr})}_{\mathrm{DDIM}})$ and initialize the truncated bridge by

$$\mathbf{X}_\varepsilon = \mathbf{X}_0 + \sigma_B s_\varepsilon \mathbf{z}_0, \qquad \mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{26}$$

Conditioned on both endpoints, the training state at ε has mean $\mathbf{X}_0 + \varepsilon(\mathbf{Y} - \mathbf{X}_0)$ and covariance $\sigma_B^2 \varepsilon(1-\varepsilon)\mathbf{I}$. Equation 26 keeps this covariance and replaces the mean by $\mathbf{X}_0$, since the target is unknown at inference. The omitted mean shift is $\varepsilon(\mathbf{Y} - \mathbf{X}_0)$, with Frobenius norm $\varepsilon\|\mathbf{Y} - \mathbf{X}_0\|_F$. We then apply Euler–Maruyama on a uniform grid from ε to $1-\varepsilon$:

$$\mathbf{X}_{t+\Delta t} = \mathbf{X}_t + \Delta t\, b_\phi(\mathbf{X}_t, t) + \sigma_B \sqrt{\Delta t}\, \mathbf{z}_t, \quad \mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{27}$$

Noise is zero on padded entries; valid states remain unbounded. After 20 steps, we evaluate the endpoint at $1-\varepsilon$ and clip it to obtain the confidence matrix. The corresponding probability-flow ODE uses v_ϕ in place of b_ϕ and removes the stochastic increments. Section B.2 gives the initialization error introduced by truncation.

B.2 Additional Derivations for the Two-Stage Solver

Structure of the lifted source. For parent-assignment matrices $\mathbf{P}_s$ and $\mathbf{P}_t$, lifting gives $\mathbf{X}_0 = \mathbf{P}_s \widehat{\mathbf{Y}}^{(\mathrm{cr})} \mathbf{P}_t^\top$, hence

$$\mathrm{rank}(\mathbf{X}_0) \le \mathrm{rank}(\widehat{\mathbf{Y}}^{(\mathrm{cr})}) \le \min(N_{\mathrm{cr}}, M_{\mathrm{cr}}). \tag{28}$$

The lifted matrix is block-constant; high-resolution refinement resolves differences within each parent block.

Orthogonal decomposition of lifting error. Let Π average a high-resolution matrix within each parent-induced block, and define $\mathcal{P}_{\mathrm{blk}} = \mathcal{U}\Pi$ as the Frobenius-orthogonal projector onto the block-constant subspace. For any compact coarse-resolution matrix $\mathbf{C}$, orthogonality gives

$$\begin{aligned} \|\mathbf{Y}^{(\mathrm{hr})} - \mathcal{U}(\mathbf{C})\|_F^2 = \|(\mathbf{I} - \mathcal{P}_{\mathrm{blk}})\mathbf{Y}^{(\mathrm{hr})}\|_F^2 \\ + \|\mathcal{P}_{\mathrm{blk}}\mathbf{Y}^{(\mathrm{hr})} - \mathcal{U}(\mathbf{C})\|_F^2. \end{aligned} \tag{29}$$

The first term contains within-block details absent from any lifted matrix. The second is the error in the block-constant component.

Endpoint-induced velocity. The endpoint prediction defines

$$v_\phi(\mathbf{X}_t, t) = \frac{\widehat{\mathbf{Y}}_t - \mathbf{X}_t}{1-t}. \tag{30}$$

For the straight-path target $u_t = \mathbf{Y}^{(\mathrm{hr})} - \mathbf{X}_{0,\mathrm{tr}}$, this gives

$$v_\phi(\mathbf{X}_t, t) - u_t = \frac{\widehat{\mathbf{Y}}_t - \mathbf{Y}^{(\mathrm{hr})}}{1-t}. \tag{31}$$

Squared velocity matching is therefore equivalent to endpoint regression with weight $(1-t)^{-2}$. We train the endpoint predictor with focal loss, which weights correspondence classification errors by prediction confidence. Equation 30 defines the sampling velocity. At $t_{K_\Pi - 1} = 1 - \Delta t$, Eq. 19 becomes

$$\mathbf{X}_{K_\Pi} = \mathrm{clip}\Big(\widehat{\mathbf{Y}}_{K_\Pi - 1}, 0, 1\Big), \tag{32}$$

The last step reaches the predicted endpoint without evaluating the denominator at $t = 1$.

Conditional Brownian drift identity and initialization. For a sampled endpoint pair and its injected noise, substituting $\widehat{\mathbf{Y}}_t = \mathbf{Y}^{(\mathrm{hr})}$ and $\widehat{\boldsymbol{\epsilon}}_t = \boldsymbol{\epsilon}$ into Eqs. 22–23 yields

$$b_\phi = \mathbf{Y}^{(\mathrm{hr})} - \mathbf{X}_{0,\mathrm{tr}} - \sigma_B \frac{t}{s_t}\boldsymbol{\epsilon} \tag{33}$$

$$= \frac{\mathbf{Y}^{(\mathrm{hr})} - \mathbf{X}_t}{1-t}, \tag{34}$$

The equality follows from $s_t^2 = t(1-t)$ and gives the endpoint-conditioned Brownian drift. At $t = \varepsilon$, the exact conditional mean is $(1-\varepsilon)\mathbf{X}_0 + \varepsilon\mathbf{Y}$. Inference omits the unavailable shift $\varepsilon(\mathbf{Y} - \mathbf{X}_0)$ while retaining the noise scale $\sigma_B s_\varepsilon$ in Eq. 26.

B.3 Matching-Matrix Endpoint Predictor

The endpoint predictor H_ϕ uses a repositioning transformer and a Sinkhorn matching head. It uses $\mathbf{C}_t = \mathbf{X}_t$ for the bounded ODE state and $\mathbf{C}_t = \mathrm{clip}(\mathbf{X}_t, 0, 1)$ for the unbounded Brownian state. After invalid entries are masked, three Sinkhorn iterations with a learned dustbin score supply weights for Soft Procrustes. The estimated transform $(\mathbf{R}_t, \boldsymbol{\tau}_t)$ warps source points to $\widetilde{\mathbf{p}}_{i,t} = \mathbf{R}_t \mathbf{p}_i + \boldsymbol{\tau}_t$, which determine the rotary positional encodings. The same $\mathbf{C}_t$ is also added to the final pairwise logits.

The high-resolution backbone features have dimension $d = 528$. A two-layer ReLU MLP processes the sinusoidal time embedding, which is added to source and target features. Six geometry-attention blocks follow in the order

$$[\mathrm{self}, \mathrm{cross}, \mathrm{self}, \mathrm{cross}, \mathrm{self}, \mathrm{cross}], \tag{35}$$

with four attention heads and rotary geometric positional encoding in each block. Let $\mathbf{h}_{s,i}$ and $\mathbf{h}_{t,j}$ denote the transformer output features after the learned matching projections, and let $\bar{\mathbf{h}} = \mathbf{h}/\sqrt{d}$. The pre-Sinkhorn endpoint logits are

$$\ell_{t,ij} = \bar{\mathbf{h}}_{s,i}^\top \bar{\mathbf{h}}_{t,j} + C_{t,ij}. \tag{36}$$

After masking invalid pairs, we apply the same three Sinkhorn iterations with a dustbin to obtain

$$\widehat{\mathbf{Y}}_t = H_\phi(\mathbf{X}_t, t, c) = \mathrm{Crop}(\mathrm{Sinkhorn}_3(\boldsymbol{\ell}_t; \alpha_{\mathrm{bin}}))\,, \tag{37}$$

Here, α_{bin} is learned and Crop removes the dustbin row and column. Mutual nearest matching with threshold 0.2 extracts correspondences from $\widehat{\mathbf{Y}}_t$. The ODE and stochastic models have separate weights but use this same architecture. The stochastic model adds a noise head to this endpoint predictor.

Noise head. The noise head shares transformer features with the endpoint head. It projects the source and target features separately and computes their pairwise inner products to capture geometric similarity. A small time-conditioned network controls how the current matching state and source matrix contribute to the prediction. These state terms are combined with the geometric similarity through a learned scale to produce signed noise estimates. Invalid pairs are masked out, and both heads update the shared transformer and backbone during training.

B.4 Training Objectives and Schedules

Implementation. Both variants use the KPFCN-style hierarchy and preprocessing from Lepard. The outputs of the deepest encoder and the first decoder provide coarse- and high-resolution features, respectively. Stage I uses deterministic DDIM ($\eta = 0$). Unless stated otherwise, both ODE and SDE use 5 Stage-I DDIM steps at the coarse resolution and 20 Stage-II bridge steps at the high resolution. The stochastic setting uses $\sigma_B = 0.1$ and $\varepsilon = 0.02$.

Each run jointly trains Stage I and one Stage-II model. Stage I predicts a clean endpoint from a noisy coarse target. Its detached prediction is lifted to initialize Stage II. Detachment blocks gradients through the source construction, including DDIM sampling, while both correspondence losses update the shared backbone.

Training supervision schedule. The source curriculum has four phases. First, a detached one-pass coarse prediction supplies the source. Second, we gradually increase the probability of using a five-step DDIM prediction. Third, the source consistently uses five-step DDIM predictions. Fourth, we vary the source by uniformly sampling a random coarse matrix or a detached DDIM prediction with one to five steps. Each coarse source is lifted to high resolution.

During the first three phases, the lifted source is mixed with the high-resolution target:

$$\mathbf{X}_{0,\mathrm{tr}} = \widetilde{\mathbf{X}}_0 = \alpha_{\mathrm{mix}} \mathbf{Y}^{(\mathrm{hr})} + (1 - \alpha_{\mathrm{mix}}) \mathbf{X}_0, \tag{38}$$

The mixing weight α_{mix} decreases linearly from 0.3 to zero and remains zero in the final phase. The endpoint supervision and branch-specific losses are maintained throughout training. Inference uses the configured DDIM budget and no ground truth.

Losses. Focal correspondence losses supervise the coarse and high-resolution endpoint predictions against their binary target matrices. Let C_{ij} be the predicted confidence clipped to $[10^{-6}, 1-10^{-6}]$. For positive entries $\mathcal{P} = \{(i,j) : Y_{ij} = 1\}$ and negative entries $\mathcal{N} = \{(i,j) : Y_{ij} = 0\}$ in the batched target tensor, we use

$$\mathcal{L}_{\mathrm{corr}}(\mathbf{C}, \mathbf{Y}) = -\alpha \left[\frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} (1 - C_{ij})^{\gamma} \log C_{ij} + \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} C_{ij}^{\gamma} \log(1 - C_{ij}) \right], \tag{39}$$

where $\alpha = 0.25$ and $\gamma = 2$, and each class mean is taken over the batch. The Sinkhorn loss excludes the dustbin; its negative-class denominator includes zero-target padding entries. Empty positive or negative sets are omitted. The motion term compares predicted and ground-truth displacements on overlapping source points. Let $\mathcal{O}$ be this source set, $\widehat{\mathbf{d}}_i = \mathbf{R}_t \mathbf{p}_i + \boldsymbol{\tau}_t - \mathbf{p}_i$, and $\mathbf{d}_i^{\star}$ the ground-truth displacement after applying the benchmark alignment to the annotated scene flow. Then

$$\mathcal{L}_{\mathrm{mot}} = \frac{1}{|\mathcal{O}|} \sum_{i\in\mathcal{O}} \|\widehat{\mathbf{d}}_i - \mathbf{d}_i^{\star}\|_1. \tag{40}$$

Following the matching-backbone implementation, we enable this term once the current batch matching recall exceeds 0.01. The shared objective is

$$\mathcal{L}_{\mathrm{shared}} = \mathcal{L}_{\mathrm{corr}}(\widehat{\mathbf{Y}}^{(\mathrm{cr})}, \mathbf{Y}^{(\mathrm{cr})}) + \mathcal{L}_{\mathrm{corr}}(\widehat{\mathbf{Y}}_t, \mathbf{Y}^{(\mathrm{hr})}) + \lambda_{\mathrm{mot}} \mathcal{L}_{\mathrm{mot}}. \tag{41}$$

For the stochastic model, let $\mathcal{P}$ and $\mathcal{N}$ denote valid positive and negative entries. Their noise errors are averaged separately to balance the sparse positives against the negative background:

$$\mathcal{L}_{\epsilon} = \frac{w_+ \mathbf{1}_{|\mathcal{P}|>0}\, \mathrm{MSE}_{\mathcal{P}} + w_- \mathbf{1}_{|\mathcal{N}|>0}\, \mathrm{MSE}_{\mathcal{N}}}{w_+ \mathbf{1}_{|\mathcal{P}|>0} + w_- \mathbf{1}_{|\mathcal{N}|>0}}, \tag{42}$$

Here, $\mathrm{MSE}_{\mathcal{P}}$ and $\mathrm{MSE}_{\mathcal{N}}$ are the mean squared errors between predicted and injected noise over the valid positive and negative entries, respectively. We use $w_+ = w_- = 0.5$; the indicators omit empty classes from the weighted average. Hard-negative noise error is logged separately and does not enter the objective. The total loss is

$$\mathcal{L} = \mathcal{L}_{\mathrm{shared}} + \mathbb{I}_{\mathrm{SDE}} \lambda_{\epsilon} \mathcal{L}_{\epsilon}. \tag{43}$$

Here, $\mathbb{I}_{\mathrm{SDE}}$ equals one for the stochastic branch and zero for endpoint-CFM. On 4DMatch, $\lambda_{\mathrm{mot}} = 0.1$ and $\lambda_{\epsilon} = 1$.

Deterministic endpoint-CFM bridge. For endpoint-CFM, we sample $t \sim \mathcal{U}(0,1)$ and interpolate between the curriculum source and ground truth. Focal correspondence loss and the optional motion loss supervise the endpoint; Eq. 30 supplies the velocity. The training objective is $\mathcal{L}_{\mathrm{shared}}$.

Stochastic paired Brownian bridge. For the stochastic model, we sample $t \sim \mathcal{U}(\varepsilon, 1-\varepsilon)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then construct $\mathbf{X}_t$ with Eq. 20. The endpoint head uses the same focal correspondence and motion losses as endpoint-CFM. The noise head receives the additional loss $\lambda_{\epsilon}\mathcal{L}_{\epsilon}$, with $\lambda_{\epsilon} = 1$.

Task-weighted field learning. Positive matches are sparse in high-resolution matrices, so averaging errors over all entries gives most of the weight to negatives. Focal loss emphasizes misclassified correspondences, and class-balanced noise regression assigns equal total weight to positive and negative entries. The identities in Sec. B.2 map the endpoint and noise predictions to velocity and drift.

Optimization details. Both variants are trained for 60 epochs with a batch size of 2, an SGD learning rate of 0.015, momentum of 0.93, weight decay of 10^{-6}, exponential decay of 0.95, and gradient clipping at 5. The Stage-I curriculum uses at most five DDIM steps. The stochastic configuration uses $w_+ = w_- = 0.5$ and $\lambda_{\epsilon} = 1$. The motion-loss weight is 0.1 for 4DMatch and zero for 3DMatch.

C Additional Experimental Details

C.1 Hierarchical Ablations

We train the stochastic coarse-only and high-resolution-only models independently, removing the inactive solver and its loss. Coarse-only lifts the Stage-I prediction for evaluation; high-resolution-only trains Stage II from an unstructured uniform source.

Table 7: Independently trained resolution ablations for the stochastic bridge on 4DLoMatch. The inactive solver and its loss are removed. Results are averaged over three sampling seeds; bold and underlined values denote the best and second-best results.

Protocol	Coarse	High-res.	NFMR↑	IR↑
Coarse-only (retrained)	✓	–	34.04	18.03
High-res.-only (retrained)	–	✓	<u>80.27</u>	<u>75.98</u>
Full two-stage	✓	✓	**82.41**	**78.79**

The full SDE design outperforms both resolution-specific variants (Table 7): NFMR/IR improve by $48.37/60.76$ points over coarse-only and $2.14/2.81$ points over high-resolution-only. These results establish the effectiveness of combining global coarse matching with fine-level bridge refinement. The following source-replacement experiments examine refinement with the trained weights fixed.

C.2 Inference-Only Source Replacement

Table 8 compares three inference settings on 4DLoMatch. The first two use the standard inference procedures of the high-resolution-only and full stochastic models. The third keeps the full-model checkpoint and initializes Stage II with an independent $\mathcal{U}(0,1)$ matrix, bypassing Stage I and lifting. All three settings use 20 SDE steps and three seeds.

Table 8: Inference-only source replacement for the stochastic bridge on 4DLoMatch. "Lifted" denotes the structured Stage-I solution, while $\mathcal{U}(0,1)$ denotes an unstructured high-resolution source. Bold and underlined values denote the best and second-best results.

Training protocol	Test-time source	NFMR↑	IR↑
High-res.-only	$\mathcal{U}(0,1)$	80.27	75.98
Full two-stage	Lifted Stage-I	<u>82.41</u>	<u>78.79</u>
Full two-stage	$\mathcal{U}(0,1)$	**82.43**	**78.92**

The full SDE model maintains high accuracy under both lifted and uniform initialization at 20 steps (Table 8). Source replacement changes NFMR/IR by only $0.02/0.13$ points, compared with sampling standard deviations of $0.24/0.22$ for the uniform source. With uniform initialization, it exceeds the high-resolution-only model by $2.16/2.94$ points. The full design therefore supports effective refinement from both structured and unstructured sources.

Source sensitivity across inference budgets. We next compare lifted and independent $\mathcal{U}(0,1)$ sources at $K \in \{1,2,5,20\}$ Stage-II steps using one fixed SDE checkpoint on 4DLoMatch. The lifted setting uses five Stage-I DDIM steps; uniform initialization bypasses Stage I. Within each pair and seed, each source is held fixed across budgets and the two settings use matched Stage-II random-number streams at each budget. We use seeds $0, 1, 2$ and matching threshold 0.2. Including the final endpoint prediction, the number of Stage-II function evaluations (NFE) is $K+1$.

Results and interpretation. With noise correction enabled, the lifted-minus-uniform NFMR/IR differences are $+0.70/+0.34$, $-0.01/-0.32$, $-0.39/-0.57$, and $-0.56/-0.67$ percentage points at 1, 2, 5, and 20 updates (Table 9). Differences are paired by input and seed before rounding. Lifting helps at one update; at larger budgets, uniform initialization matches or exceeds its accuracy. Both sources peak at two updates.

Table 9: Source and noise-correction diagnostics across SDE budgets on 4DLoMatch. Values are mean ± population standard deviation over three sampling seeds. The noise-correction ablation sets the predicted noise to zero during inference while retaining stochastic increments and the final endpoint prediction. Bold marks the higher mean between sources within each correction setting and budget.

			With noise correction		Without noise correction	
Steps	NFE	Source	NFMR (%) ↑	IR (%) ↑	NFMR (%) ↑	IR (%) ↑
1	2	Lifted	$\mathbf{82.79} \pm 0.01$	$\mathbf{79.34} \pm 0.03$	$\mathbf{82.50} \pm 0.02$	$\mathbf{79.17} \pm 0.04$
1	2	Uniform	82.10 ± 0.15	79.00 ± 0.11	81.73 ± 0.29	78.70 ± 0.23
2	3	Lifted	83.06 ± 0.15	79.45 ± 0.18	$\mathbf{81.98} \pm 0.17$	$\mathbf{78.56} \pm 0.19$
2	3	Uniform	$\mathbf{83.07} \pm 0.22$	$\mathbf{79.77} \pm 0.18$	81.02 ± 0.13	77.79 ± 0.06
5	6	Lifted	82.15 ± 0.07	78.53 ± 0.07	$\mathbf{81.74} \pm 0.06$	$\mathbf{78.29} \pm 0.06$
5	6	Uniform	$\mathbf{82.54} \pm 0.13$	$\mathbf{79.10} \pm 0.10$	81.19 ± 0.15	77.90 ± 0.05
20	21	Lifted	82.00 ± 0.06	78.40 ± 0.04	$\mathbf{81.65} \pm 0.17$	$\mathbf{78.16} \pm 0.11$
20	21	Uniform	$\mathbf{82.56} \pm 0.15$	$\mathbf{79.07} \pm 0.17$	81.36 ± 0.21	78.03 ± 0.18

The source sweep uses a fixed checkpoint; the resolution ablations in Table 7 use separately trained models. Runtime for lifted initialization includes the Stage-I computation.

Effect of noise correction. The learned noise correction improves SDE refinement for both sources at every tested budget. At two lifted-source updates, it raises NFMR/IR by $1.08/0.89$ points. The ablation sets the predicted noise to zero, reducing the drift to $\hat{\mathbf{Y}}_t - \mathbf{X}_0$, while preserving the initial perturbation, Brownian increments, and final endpoint prediction.

With the correction disabled, lifted-source NFMR falls from 82.50 at one step to 81.65 at twenty steps. Without correction, lifted initialization outperforms uniform initialization at every tested budget. The multi-step decrease occurs with and without noise correction; the relative performance of the two sources changes with the update rule.

Source replacement in endpoint-CFM. The endpoint-CFM sweep uses one checkpoint trained with the lifted-source protocol. Both settings run the five-step Stage-I solver with matched coarse random-number streams; in the uniform setting, only the Stage-II source is replaced by $\mathcal{U}(0, 1)$. We verify equal coarse predictions across paired cases and keep each source fixed across budgets. Seeds are $0, 1, 2$, the matching threshold is 0.2, and the weights are unchanged. Euler integration is deterministic, so seed variation comes from initialization. The ODE solver uses K Stage-II function evaluations for K integration steps.

Table 10: Inference-time source replacement for one endpoint-CFM model trained with the lifted-source protocol on 4DLoMatch. Both settings share the same checkpoint. Values are mean ± population standard deviation across three initialization seeds. Bold indicates the higher mean within each budget. Stage-II NFE equals the number of steps.

Steps / NFE	Source	NFMR (%) ↑	IR (%) ↑
1	Lifted	$\mathbf{75.82} \pm 0.10$	$\mathbf{70.07} \pm 0.16$
1	Uniform	70.95 ± 0.07	65.68 ± 0.06
2	Lifted	76.32 ± 0.01	70.04 ± 0.07
2	Uniform	$\mathbf{76.76} \pm 0.04$	$\mathbf{70.71} \pm 0.05$
5	Lifted	77.80 ± 0.06	71.05 ± 0.04
5	Uniform	$\mathbf{78.04} \pm 0.12$	$\mathbf{71.59} \pm 0.12$
20	Lifted	78.44 ± 0.10	71.57 ± 0.10
20	Uniform	$\mathbf{78.57} \pm 0.11$	$\mathbf{71.86} \pm 0.11$

Inference-time initialization sensitivity. For endpoint-CFM, the lifted-minus-uniform NFMR/IR differences are $+4.87/+4.40$, $-0.44/-0.68$, $-0.23/-0.54$, and $-0.12/-0.29$ percentage points at 1, 2, 5, and 20 updates (Table 10), computed before rounding. The lifted source helps most at one update. Uniform initialization gives slightly higher means with two or more updates, and both sources improve from two to five to twenty updates.

Noise correction and solver comparison. The SDE design achieves $82.00/78.40$ NFMR/IR at twenty lifted-source updates, compared with $78.44/71.57$ for endpoint-CFM. Its learned noise correction improves the SDE result from $81.65/78.16$ to $82.00/78.40$. The independently trained ODE and SDE models implement their respective paths, prediction heads, losses, and inference rules; the correction ablation evaluates the SDE drift within the same model.

Training and inference results. Together, the resolution and source experiments demonstrate the effectiveness of the full SDE design, which maintains high matching accuracy after 20 updates with either lifted or uniform initialization. It improves NFMR/IR over the high-resolution-only variant from $80.27/75.98$ to $82.41/78.79$. Lifting benefits one-update inference, while longer refinement achieves strong results from either source. Section C.3 evaluates the additional recovery enabled by retaining all candidates.

C.3 Inference-Only Top-K Candidate Diagnostic

Protocol. This test restricts candidates at inference using one SDE checkpoint trained with all candidates. On 4DLoMatch, we compare full-matrix inference (FULL), an all-candidate mask (ALL), and global coarse-pair Top-K masks for $K \in \{32, 64, 128, 256\}$. We rank all valid coarse pairs by Stage-I score and break ties by flattened index. The selected pairs are lifted to a fixed fine-level mask. Restricted runs apply this mask to the source, stochastic updates, and matching logits, while keeping the dustbin. All settings use five coarse steps, twenty fine steps (21 Stage-II NFE), threshold 0.2, and seeds $0, 1, 2$. Inputs, coarse predictions, and stochastic streams are paired. Selection uses predicted coarse scores. The learned weights remain fixed across settings.

GT coverage and FULL recovery. Let G denote the set of supervised fine-level ground-truth pairs, M_K the lifted candidate mask, and P_{FULL} the discrete correspondences predicted by unrestricted inference. We measure

$$\mathrm{Coverage}(K) = \frac{|G \cap M_K|}{|G|}, \qquad \mathrm{Recovery}(K) = \frac{|P_{\mathrm{FULL}} \cap (G \setminus M_K)|}{|G \setminus M_K|}.$$

Recovery counts exact supervised pairs predicted by FULL outside the mask; NFMR/IR instead use geometric tolerance. For each seed, coverage and recovery are calculated from counts summed over test pairs, then averaged across seeds. NFMR/IR are averaged over pairs and seeds. Recovery is undefined when the mask excludes no ground-truth pairs.

Table 11: Candidate exclusion and recovery on 4DLoMatch for one fixed checkpoint trained with full candidates. Masks are applied at inference. All values are percentages except the FULL-minus-restricted NFMR difference in percentage points (pp). Results average three sampling seeds. FULL recovery is the fraction of excluded GT pairs predicted correctly by unrestricted inference.

Mode	NFMR ↑	IR ↑	GT coverage	FULL recovery	ΔNFMR (pp)
FULL	82.04	78.46	100.00	—	0.00
ALL	82.04	78.46	100.00	—	0.00
Top-32	18.19	42.88	7.46	52.66	63.85
Top-64	28.15	40.78	13.03	51.91	53.89
Top-128	39.26	39.04	21.83	50.78	42.78
Top-256	48.94	39.26	34.23	49.30	33.10

Results. Retaining the full matching space substantially improves correspondence recovery (Table 11). FULL and ALL give identical scores. Top-256 retains 34.23% of supervised pairs, while FULL recovers 49.30% of its excluded pairs and raises NFMR by 33.10 points. Recovery remains

50.78–52.66% for the smaller masks. These results show how full-candidate refinement corrects coarse selection errors by retaining access to valid fine-level matches.

All settings use dense computation and the same full-candidate-trained weights. The masks determine which candidates participate in inference.

C.4 Evaluation Protocol Details

We follow the Lepard protocol Li & Harada (2022a) on 4DMatch/4DLoMatch and the filtered 4DMatch-F/4DLoMatch-F splits used by NDP and GraphSCNet Li & Harada (2022b); Qin et al. (2023) for downstream registration. At a $4\,\mathrm{cm}$ threshold, NFMR evaluates warped metric points obtained by three-neighbor interpolation from correspondence anchors. IR measures the matches consistent with ground-truth scene flow. For CAPE and DeepDeform, absolute and relative errors are $e_i = \|\widehat{\mathbf{f}}_i - \mathbf{f}_i\|_2$ and $r_i = e_i / \max(\|\mathbf{f}_i\|_2, \epsilon)$. EPE averages e_i in meters. AccS/AccR count points below either the absolute or relative threshold: $(2\,\mathrm{cm}, 5\%)/(5\,\mathrm{cm}, 10\%)$ for CAPE and $(2.5\,\mathrm{cm}, 2.5\%)/(5\,\mathrm{cm}, 5\%)$ for DeepDeform. OR counts points with $r_i > 30\%$. All metrics are averaged over pairs; all except EPE are percentages. Lower EPE/OR and higher NFMR/IR/AccS/AccR are better. Bold and underlined entries mark the best and second-best values within the stated comparison group.

MPC-CAPE and MPC-DD are the public preprocessed CAPE and DeepDeform datasets released with SyNoRiM Huang et al. (2022). Following RGGT Zheng et al. (2026), we expand each of the 209 four-frame MPC-CAPE samples into 12 ordered source–target pairs, giving 2,508 pairs. Diff-Reg and our models use 4DMatch checkpoints and the same pretrained GraphSCNet solver, without CAPE adaptation. The deformation-stratified evaluation reuses these checkpoints without retraining.

For MPC-DD, we fit a least-squares rigid transform to points marked visible by the flow mask, following GraphSCNet Qin et al. (2023). We evaluate the mean residual over the complete source cloud and remove pairs below 1 cm. Public MPC-DD preprocessing leaves 1,221 pairs, distinct from GraphSCNet's 1,011-pair split. The re-evaluated zero-shot models and their GraphSCNet solver are trained on 4DMatch, with no DeepDeform fine-tuning. EPE is averaged over all source points and then over pairs. SyNoRiM Huang et al. (2022) supplies a target-domain-trained reference.

C.5 Baseline Training and Comparison Context

Result sources. The 4DMatch/4DLoMatch correspondence and 4DMatch-F/4DLoMatch-F registration baselines are taken from the cited literature. Baseline rows labeled GraphSCNet report published results using this deformation solver. On CAPE, we evaluate Diff-Reg and our models and quote other scores from RGGT Zheng et al. (2026). On DeepDeform, SyNoRiM, Lepard, Diff-Reg, and our models are evaluated on the same 1,221 pairs. The cited AniSym-Net and GeoTransformer+GraphSCNet results use different evaluation protocols.

RGGT Zheng et al. (2026) provides two 4DMatch-F settings. Compared with RGGT without TRELLIS features, the stochastic bridge lowers EPE by 0.003, raises AccS/AccR by $12.0/7.4$ points, and lowers the outlier rate by 3.9 points. With TRELLIS features, RGGT has lower EPE and outlier rate and higher AccR, while our stochastic model has slightly higher AccS (74.1 versus 73.9). RGGT without TRELLIS features does not report 4DLoMatch-F results.

For CAPE, the cited SyNoRiM Huang et al. (2022) result uses target-domain self-supervision and four-view synchronization. GeoTransformer Qin et al. (2022) is trained on ModelNet, and GraphSCNet Qin et al. (2023) on 4DMatch. Diff-Reg and our models use 4DMatch checkpoints with pairwise input and no target-domain adaptation.

DeepDeform references. Re-evaluated on the same pairs with each directed pair supplied as a two-view sequence, the MPC-DD-trained SyNoRiM reaches 0.0268 EPE, 72.9 AccS, 91.8 AccR, and 12.8 OR. AniSym-Net Wang et al. (2025) reports $2.66\,\mathrm{cm}$ EPE and 89.43% AccR in its Table IV after MPI-FAUST-to-DeepDeform fine-tuning. These correspond to $0.0266\,\mathrm{m}$ and 89.4% after rounding; AccS and OR are not reported for this setting. AniSym-Net and the published GeoTransformer+GraphSCNet result use different evaluation protocols and are listed separately from the matched zero-shot results.

Table 12: Zero-shot CAPE performance grouped by mean ground-truth deformation magnitude. EPE is in meters; other metrics are percentages. The groups use shared test-set tertile boundaries and contain 836 pairs each; stochastic results are averaged over three sampling seeds. Best and second-best values are determined separately within each deformation subset.

Method	Deformation	EPE (m)↓	AccS↑	AccR↑	OR↓
Diff-Reg	Small	0.012240	87.304	**97.682**	<u>64.826</u>
	Medium	0.017558	78.749	95.285	42.121
	Large	0.024673	69.102	92.249	29.312
Ours^{ODE}	Small	<u>0.011739</u>	<u>87.903</u>	<u>97.534</u>	65.754
	Medium	<u>0.016726</u>	<u>79.191</u>	<u>95.419</u>	<u>41.136</u>
	Large	<u>0.022123</u>	<u>70.466</u>	<u>93.025</u>	<u>27.332</u>
$\text{Ours}^{\text{Stochastic}}$	Small	**0.011418**	**88.459**	**97.682**	**64.170**
	Medium	**0.016416**	**80.369**	**95.827**	**39.982**
	Large	**0.021271**	**72.417**	**93.681**	**26.081**

C.6 CAPE Results by Deformation Magnitude

We sort the 2,508 CAPE pairs by mean ground-truth flow magnitude and divide them into three equal groups. The same tertile boundaries are used for every method.

Analysis of deformation severity. The SDE bridge leads the evaluated methods across deformation levels (Table 12), with the lowest EPE and OR and the highest AccS in all three groups. AccR is highest for medium and large deformations and tied for highest for small deformations at the reported precision. On the large-deformation subset, SDE reaches 0.021271 EPE and 72.417% AccS, compared with ODE's 0.022123 and 70.466%. The consistent results across shared motion groups demonstrate effective generalization across deformation magnitudes. OR is compared within each group because its relative-error denominator depends on motion magnitude.

C.7 Zero-Shot Transfer to Complete Shapes on TOSCA

We test whether both variants trained on partial scans in 4DMatch transfer to complete-shape correspondence on TOSCA, without training on SMAL or TOSCA. We use 41 animal shapes and all 286 within-category directed, non-self pairs under the DPC/SE-ORNet reference protocol. Each shape is represented by 1,024 points sampled from its complete surface. Corresponding vertex subsets are fixed during benchmark preparation, and coordinates are divided by 100. Ground-truth maps are used only for data preparation and evaluation. For dense prediction, each input source point inherits the row-maximum match of its nearest high-resolution source anchor, and the matched target anchor is mapped to the nearest input target point. No GraphSCNet or additional test-time optimization is used. Acc@1% counts predictions whose Euclidean error is below 1% of the sampled target's maximum pairwise distance, following Chen et al. (2026). Both variants use their epoch-30 checkpoints and the same test inputs and decoder. ODE uses 10 coarse and 20 fine steps; the stochastic model uses 5 coarse and 20 fine steps.

Both bridge designs extend from partial-scan registration to complete-shape correspondence without target-domain training or additional test-time optimization. SDE reaches 54.01% accuracy, exceeding ODE's 49.51% by 4.50 points (Table 13). These results demonstrate that models trained on partial scans can also match complete shapes.

Our models use supervised 4DMatch training; the reference methods use unsupervised SMAL training and their published evaluation protocols.

C.8 Solver-Step Sensitivity

Table 14 studies the iteration budgets of both Stage-II models. Let K_{I} denote the number of Stage-I DDIM steps at the coarse resolution, and let K_{II} denote the number of Stage-II Euler or Euler–Maruyama steps at the high resolution. These inference-only tests reuse fixed checkpoints without retraining. We first vary $K_{\mathrm{I}} \in \{1, 3, 5, 10, 20\}$ while fixing $K_{\mathrm{II}} = 20$. We then vary $K_{\mathrm{II}} \in$

Table 13: TOSCA correspondence accuracy, grouped by additional test-time optimization. SMAL-trained values are from Table 1 of Chen et al. (2026); all rows exclude TOSCA training. Ours reports pair-averaged mean ± population standard deviation over three inference seeds. Training data and test-time optimization are specified for each method.

Method	Venue	Training data	Additional optimization	Acc@1% (%)↑
With additional test-time optimization				
EquiShape+	TIP 2025	SMAL	Yes	57.7
Chen et al.+	CVPR 2026	SMAL	Yes	60.5
Without additional test-time optimization				
CorrNet3D	CVPR 2021	SMAL	No	5.3
DPC	3DV 2021	SMAL	No	33.2
SE-ORNet	CVPR 2023	SMAL	No	36.4
HSTR	TIP 2023	SMAL	No	33.9
TANet	CVPR 2024	SMAL	No	37.1
AMIGO	AAAI 2025	SMAL	No	38.5
DiffCorr	AAAI 2025	SMAL	No	41.6 ± 0.6
DV-Matcher	CVPR 2025	SMAL	No	39.5
Chen et al.	CVPR 2026	SMAL	No	54.0
Ours^{ODE}	–	4DMatch	No	49.51 ± 0.02
Ours^{Stochastic}	–	4DMatch	No	54.01 ± 0.02

Table 14: Shared solver-step sensitivity on 4DLoMatch. The Stage-I sweep fixes $K_{\mathrm{II}} = 20$, and the Stage-II sweep fixes $K_{\mathrm{I}} = 5$. Brownian-bridge results are mean±standard deviation over three sampling seeds. T_{I} and T_{II} are measured separately per pair on the same hardware. For correspondence metrics, bold and underlined values denote the best and second-best settings within each sweep and branch; latency and memory are not ranked.

			Ours^{ODE}		Ours^{Stochastic}		Latency (ms)↓			Memory (GiB)↓	
Sweep	K_{I}	K_{II}	NFMR↑	IR↑	NFMR↑	IR↑	$T_{\mathrm{I}}^{\mathrm{ODE}}/T_{\mathrm{I}}^{\mathrm{SDE}}$	$T_{\mathrm{II}}^{\mathrm{ODE}}$	$T_{\mathrm{II}}^{\mathrm{SDE}}$	ODE	SDE
Stage I	1	20	78.27	71.42	$\underline{82.12 \pm 0.08}$	78.47 ± 0.08	50/58	578	727	1.86	1.86
	3	20	78.36	71.52	82.03 ± 0.03	78.40 ± 0.06	114/96	748	664	1.85	1.86
	5	20	$\underline{78.46}$	$\underline{71.67}$	82.04 ± 0.08	78.46 ± 0.09	162/133	740	639	1.85	1.86
	10	20	78.37	71.51	$\mathbf{82.19 \pm 0.20}$	$\mathbf{78.58 \pm 0.21}$	283/194	727	503	1.86	1.86
	20	20	**78.68**	**71.80**	$\underline{82.12 \pm 0.04}$	$\underline{78.54 \pm 0.02}$	513/308	713	426	1.85	1.86
Stage II	5	1	75.76	70.11	$\underline{82.94 \pm 0.10}$	$\underline{79.52 \pm 0.04}$	162/99	46	44	1.85	1.85
	5	2	76.38	70.11	$\mathbf{83.16 \pm 0.13}$	$\mathbf{79.62 \pm 0.11}$	162/98	82	64	1.86	1.85
	5	5	$\underline{77.98}$	71.23	82.29 ± 0.03	78.73 ± 0.03	159/96	189	125	1.85	1.85
	5	10	**78.46**	**71.71**	82.22 ± 0.11	78.61 ± 0.08	152/94	355	225	1.85	1.85
	5	20	**78.46**	$\underline{71.67}$	82.04 ± 0.08	78.46 ± 0.09	162/133	740	639	1.85	1.86
	5	40	**78.46**	71.63	82.03 ± 0.21	78.44 ± 0.18	150/89	1380	814	1.85	1.86

$\{1, 2, 5, 10, 20, 40\}$ while fixing $K_{\mathrm{I}} = 5$. We evaluate endpoint-CFM once for each pair and average the Brownian-bridge results over three seeds.

Runtime and memory protocol. We measure synchronized wall-clock latency on an NVIDIA RTX 4090 GPU with a batch size of one and average it over all test pairs. The backbone features are computed once for each pair and then reused, while the matching transformer and prediction heads run at every step. T_{I} includes backbone encoding, multiresolution preprocessing, coarse DDIM inference, and hierarchical lifting. T_{II} includes high-resolution bridge integration, correspondence extraction, and Soft Procrustes. Both measurements exclude data loading, metric computation, and downstream RANSAC. Backbone latency is included in T_{I}. Peak GPU memory includes model parameters and intermediate tensors but excludes the reserved CUDA cache. We average the stochastic latency across seeds and report the largest memory peak.

A small coarse-stage budget already supports accurate matching (Table 14). Increasing Stage-I steps from 1 to 20 changes ODE NFMR/IR by only $0.41/0.38$ points and SDE by $0.00/0.07$ points. The fine-stage sweep below shows how each bridge uses its refinement budget.

ODE iterative refinement improves matching beyond the single-step endpoint prediction. One Euler step at $t = 0$ reaches the clipped endpoint prediction; ten steps raise NFMR/IR from $75.76/70.11$ to $78.46/71.71$ ($+2.70/+1.60$ points), after which accuracy plateaus. The stochastic model improves from $82.94/79.52$ at one update to $83.16/79.62$ at two ($+0.22/+0.10$ points), then loses accuracy at larger budgets. Each stochastic run includes a final endpoint prediction: K updates cost $K + 1$ network evaluations, compared with K for endpoint-CFM. Both models use twenty updates by default.

Runtime and memory. The measured costs demonstrate the practical value of reduced-step refinement. Ten ODE steps take 355 ms in Stage II, versus 740 ms at twenty, with NFMR/IR of $78.46/71.71$ versus $78.46/71.67$: a $2.08\times$ Stage-II speedup. Two SDE updates take 64 ms versus 639 ms at twenty, a $9.98\times$ Stage-II speedup, while improving NFMR/IR from $82.04/78.46$ to $83.16/79.62$. Peak memory stays within 1.85–1.86 GiB across the sweep. Table 14 reports Stage-I costs separately.

C.9 Inference Controls on a Fixed SDE Checkpoint

Protocol. We evaluate one frozen SDE-trained checkpoint on 4DLoMatch. The lifted source uses five Stage-I DDIM updates; the uniform source draws an independent $\mathcal{U}(0, 1)$ matrix and bypasses Stage I. Within each source, all modes and budgets use the same input and initial state $\mathbf{X}_\varepsilon = \mathbf{X}_0 + \sigma_B \sqrt{\varepsilon(1-\varepsilon)}\,\mathbf{z}$, with $\sigma_B = 0.1$ and $\varepsilon = 0.02$. Correspondences use the same mutual matching rule and confidence threshold 0.2. The subsequent updates are deterministic conditional on this initial state.

Update rules. The forward-drift mode retains the endpoint and noise heads and integrates $\mathbf{X}_{k+1} = \mathbf{X}_k + \Delta t\, b_\phi(\mathbf{X}_k, t_k)$, using the complete learned drift in Eq. 23. It removes only the Brownian increment from Eq. 27. The probability-flow mode instead integrates $\mathbf{X}_{k+1} = \mathbf{X}_k + \Delta t\, v_\phi(\mathbf{X}_k, t_k)$ using Eq. 22. Both run K updates over $[\varepsilon, 1-\varepsilon]$ and read the endpoint once more, costing $K + 1$ Stage-II network evaluations. Direct endpoint prediction queries the endpoint head once at $(\mathbf{X}_\varepsilon, \varepsilon)$ and uses its output for matching. All three modes use the same weights; the probability-flow mode here differs from the separately trained endpoint-CFM model. Code names are `drift_no_noise`, `pf_ode`, and `endpoint`, respectively.

Table 15: Full inference-control sweep on one SDE checkpoint, 4DLoMatch. NFMR and IR are percentages. Initial perturbations are paired within each source; neither update mode injects subsequent random increments. Direct endpoint prediction is evaluated once per source, at 1 NFE.

Source	Inference mode	Updates	Stage-II NFE	NFMR↑	IR↑
Lifted	Forward drift	1	2	83.94	80.49
Lifted	Forward drift	2	3	83.44	79.84
Lifted	Forward drift	5	6	82.62	79.03
Lifted	Forward drift	20	21	82.37	78.70
Lifted	Probability flow	1	2	74.06	72.56
Lifted	Probability flow	2	3	75.32	73.86
Lifted	Probability flow	5	6	78.05	75.97
Lifted	Probability flow	20	21	80.04	76.89
Lifted	Direct endpoint	0	1	79.28	76.69
Uniform	Forward drift	1	2	82.82	79.61
Uniform	Forward drift	2	3	82.56	79.33
Uniform	Forward drift	5	6	83.16	79.64
Uniform	Forward drift	20	21	82.63	79.12
Uniform	Probability flow	1	2	74.85	73.02
Uniform	Probability flow	2	3	74.76	72.96
Uniform	Probability flow	5	6	76.42	74.42
Uniform	Probability flow	20	21	80.01	77.41
Uniform	Direct endpoint	0	1	75.12	72.79

Target-directed contraction and learned-drift error. The endpoint-conditioned drift in Eq. 34 defines the reference field $b^*(\mathbf{X}, t) = (\mathbf{Y} - \mathbf{X})/(1 - t)$. An Euler update with this field and step size $0 < h \leq 1 - t$ contracts the target error:

$$\mathbf{X}_{t+h}^{*,\mathrm{Euler}} - \mathbf{Y} = \left(1 - \frac{h}{1-t}\right)(\mathbf{X}_t - \mathbf{Y}). \tag{44}$$

For the implemented initialization $\mathbf{X}_\varepsilon = \mathbf{X}_0 + \sigma_B s_\varepsilon \mathbf{z}$, define $\Delta b_\varepsilon = b_\phi(\mathbf{X}_\varepsilon, \varepsilon) - (\mathbf{Y} - \mathbf{X}_\varepsilon)/(1-\varepsilon)$. The single update $\mathbf{X}^+ = \mathbf{X}_\varepsilon + (1 - 2\varepsilon) b_\phi(\mathbf{X}_\varepsilon, \varepsilon)$ then satisfies

$$\mathbf{X}^+ - \mathbf{Y} = \frac{\varepsilon}{1-\varepsilon}(\mathbf{X}_\varepsilon - \mathbf{Y}) + (1 - 2\varepsilon)\Delta b_\varepsilon. \tag{45}$$

This exact decomposition separates initial-state contraction from the learned-drift residual under the implemented initialization. For $\Delta b_\varepsilon = 0$, the ideal contraction factor is $\varepsilon/(1-\varepsilon) \approx 0.0204$ at $\varepsilon = 0.02$. The equation describes the state before the final endpoint prediction at $1 - \varepsilon$, which completes the two-NFE procedure evaluated below.

Few-step SDE refinement. With the lifted source, one forward-drift update followed by endpoint prediction reaches $83.94/80.49$ NFMR/IR, compared with $79.28/76.69$ for direct endpoint prediction. The gains are $4.66/3.80$ percentage points. With the uniform source, the corresponding gain is $7.70/6.82$ points, from $75.12/72.79$ to $82.82/79.61$. These gains demonstrate the effectiveness of SDE iterative refinement with just one forward-drift update. On 4DMatch, the same lifted-source configuration reaches $93.10/92.25$ NFMR/IR at 2 Stage-II NFE, improving over the standard stochastic result by $0.67/0.78$ points (Table 1).

Forward drift and probability flow. Forward-drift updates outperform probability-flow updates at every matched source and NFE budget in Table 15. At 2 NFE, the lifted-source gap is $9.88/7.93$ NFMR/IR points. The gap narrows to $2.33/1.81$ points at 21 NFE as probability-flow accuracy increases. Uniform-source probability flow also improves overall with more updates, reaching $80.01/77.41$ at 21 NFE. The consistent advantage across sources and budgets shows that the SDE bridge learns a forward drift well suited to correspondence refinement. Its largest gains occur with one update and two Stage-II network evaluations.

Initialization and budget. The lifted source is particularly effective for few-step SDE refinement. It improves direct endpoint NFMR/IR over uniform initialization by $4.16/3.90$ points and retains a $1.12/0.88$-point advantage after one forward-drift update. Lifted-source refinement reaches its highest accuracy at this single update; the uniform source peaks at five updates with $83.16/79.64$ and exceeds lifting at five and twenty updates. Together, these results demonstrate rapid refinement from the structured source and effective iterative recovery from an unstructured source.

D Limitations and Future Work

Deterministic transport. Endpoint-CFM uses fixed coupling and straight training paths. Reflow or trajectory distillation may reduce the number of inference steps needed to reach a given matching accuracy.

Trajectory learning. Training samples analytic ground-truth-conditioned states, whereas inference follows predicted states. Extending the source curriculum to include predicted trajectories could reduce accumulated errors over long inference paths.

E FAILURE CASES

We examine the three pairs with the highest EPE in the displayed stochastic run on each dataset, restricted to pairs evaluated by all three methods. Figures 4 and 5 show ambiguous correspondences between similar body regions under partial visibility.

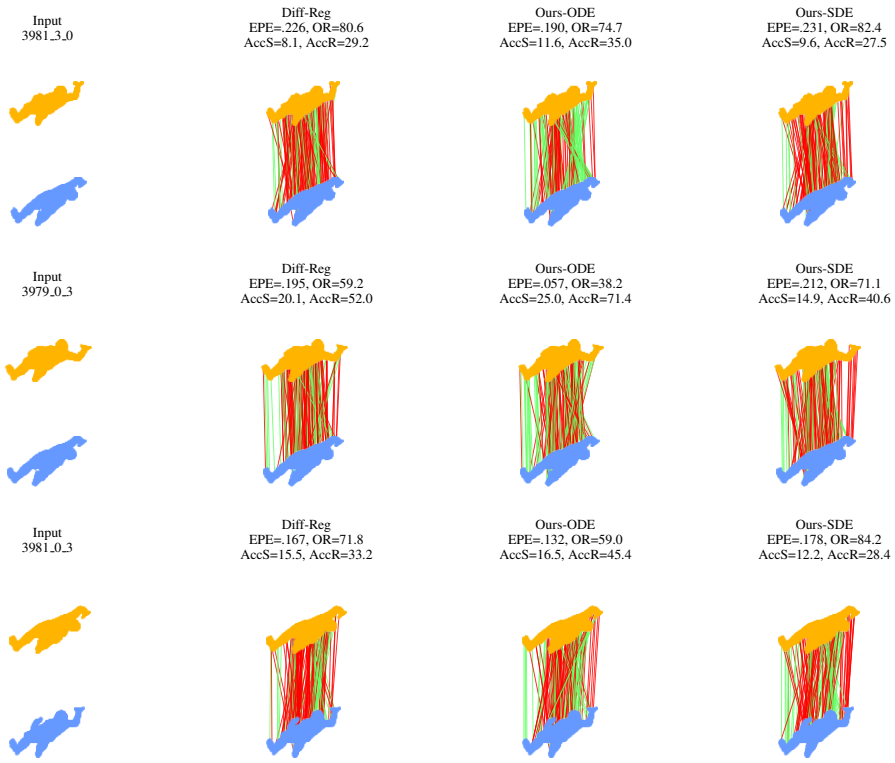


Figure 4: CAPE failure cases. The three common test pairs with the highest EPE in the displayed Ours-SDE run are shown in descending order. Each row contains the separated input point clouds followed by correspondences from Diff-Reg, Ours-ODE, and Ours-SDE. Orange and blue denote the source and target, while light-green and red lines denote correct and incorrect correspondences, respectively. EPE is measured in meters; AccS, AccR, and OR are percentages. The separation between the source and target is introduced only for visualization.

In the CAPE examples (Fig. 4), the lying poses make body regions difficult to distinguish from overall shape. Limb bends and visible surface details provide local cues, but all three methods produce incorrect matches. The stochastic EPE values are 0.231, 0.212, and 0.178, compared with 0.190, 0.057, and 0.132 for endpoint-CFM. Diff-Reg also has lower EPE than the stochastic model on these three pairs.

The DeepDeform examples contain partially visible bodies and similarly shaped local regions (Fig. 5). On the first pair, endpoint-CFM has lower EPE (0.146) than Diff-Reg (0.838) and the stochastic model (0.875), yet its AccS and AccR are zero and OR is 100%. The stochastic model has EPEs of 0.875, 0.648, and 0.608 across the three pairs, with zero AccS/AccR and 100% OR throughout. All three methods fail on the first pair.

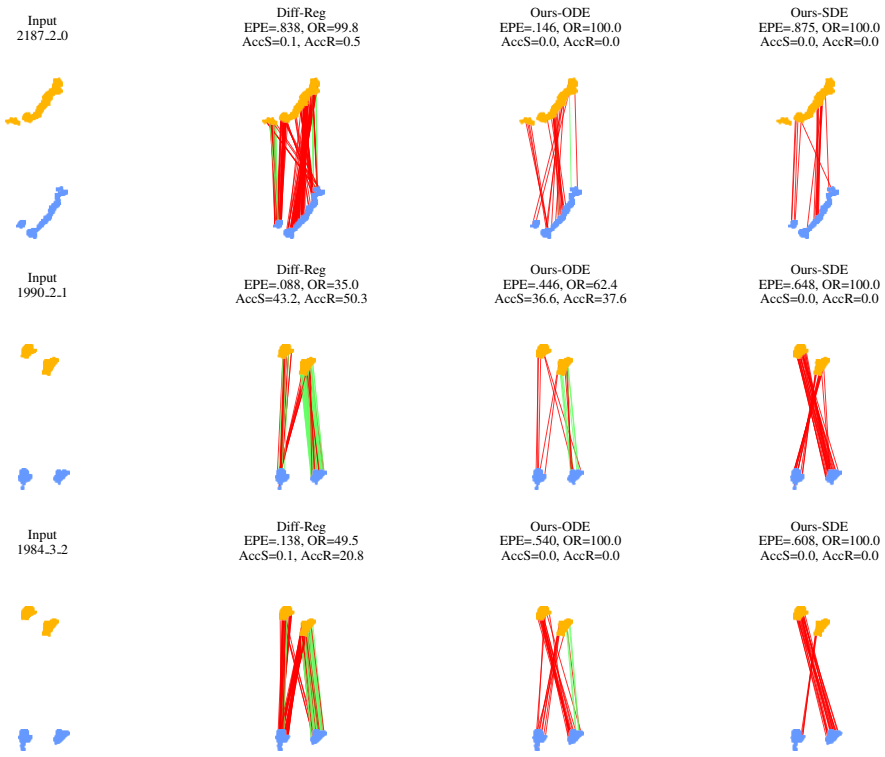
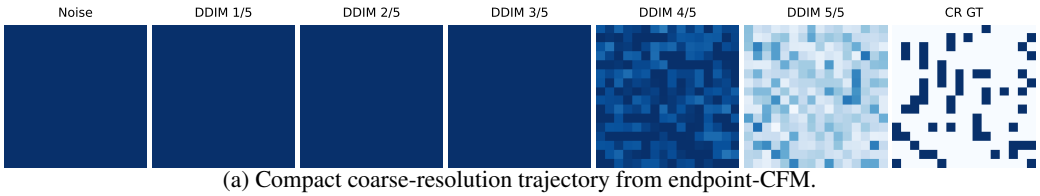


Figure 5: DeepDeform failure cases under the same protocol as Fig. 4. The rows are the three common test pairs with the highest EPE in the displayed Ours-SDE run. Colors, correspondence correctness, metrics, and the visualization-only separation follow Fig. 4.

F MATCHING-MATRIX TRAJECTORY VISUALIZATION

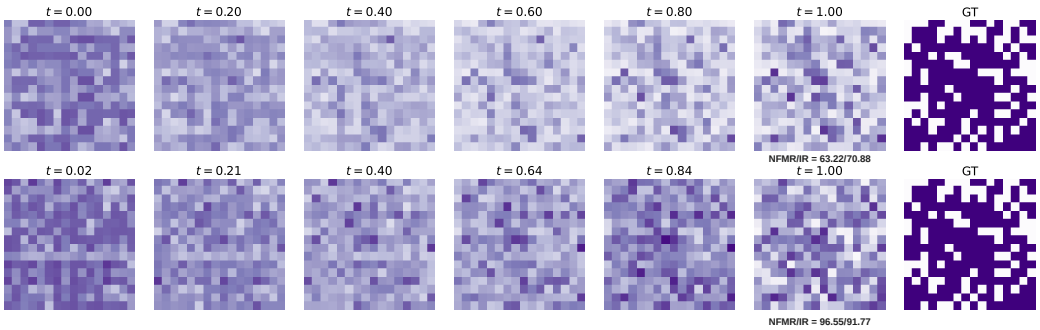
Figure 6 visualizes example 00105, on which the stochastic model achieves higher correspondence scores than endpoint-CFM. The coarse panels show the initial state and five DDIM updates; the fine panel shows the Stage-II trajectories. The visible differences are concentrated in Stage II. Matrices are pooled only for display; inference uses the original resolution.



(a) Compact coarse-resolution trajectory from endpoint-CFM.



(b) Compact coarse-resolution trajectory from the stochastic bridge.



(c) Index 00105: high-resolution Stage-II trajectories of endpoint-CFM (top) and the stochastic bridge (bottom).

Figure 6: Matching-matrix trajectories on 4DLoMatch. (a)–(b) show the initial state and five Stage-I DDIM updates for index 00105, in the compact coarse-resolution space. (c) compares its deterministic ODE and stochastic SDE trajectories in the high-resolution space; endpoint-CFM obtains 63.22 NFMR and 70.88 IR, and the stochastic bridge obtains 96.55 NFMR and 91.77 IR. Matrices are max-pooled into 16×16 mosaics for visualization only; the final column of each panel is the corresponding ground-truth matching matrix.