Accurate Point Cloud Registration with Robust Optimal Transport

Zhengyang Shen[*]
UNC Chapel Hill
zyshen@cs.unc.edu

Jean Feydy[*]
Imperial College London
jfeydy@ic.ac.uk

Peirong Liu
UNC Chapel Hill
peirong@cs.unc.edu

Ariel Hernán Curiale
Harvard Medical School
acuriale@bwh.harvard.edu

Ruben San José Estépar
Harvard Medical School
rubensanjose@bwh.harvard.edu

Raúl San José Estépar
Harvard Medical School
rjosest@bwh.harvard.edu

Marc Niethammer
UNC Chapel Hill
mn@cs.unc.edu

Abstract

This work investigates the use of robust optimal transport (OT) for shape matching. Specifically, we show that recent OT solvers improve both optimization-based and deep learning methods for point cloud registration, boosting accuracy at an affordable computational cost. This manuscript starts with a practical overview of modern OT theory. We then provide solutions to the main difficulties in using this framework for shape matching. Finally, we showcase the performance of transport-enhanced registration models on a wide range of challenging tasks: rigid registration for partial shapes; scene flow estimation on the Kitti dataset; and nonparametric registration of lung vascular trees between inspiration and expiration. Our OT-based methods achieve state-of-the-art results on Kitti and for the challenging lung registration task, both in terms of accuracy and scalability.
We also release PVT1010, a new public dataset of 1,010 pairs of lung vascular trees with densely sampled points. This dataset provides a challenging use case for point cloud registration algorithms with highly complex shapes and deformations. Our work demonstrates that robust OT enables fast pre-alignment and fine-tuning for a wide range of registration models, thereby providing a new key method for the computer vision toolbox. Our code and dataset are available online at: https://github.com/uncbiag/robot.

1 Introduction

Shape registration is a fundamental but difficult problem in computer vision. The task is to determine plausible spatial correspondences between pairs of shapes, with use cases that range from pose estimation for noisy point clouds [14] to the nonparametric registration of high-resolution medical images [17]. As illustrated in Fig. 1, most existing approaches to this problem consist of a combination of three steps, possibly fused together by some deep learning (DL) methods: (1) feature extraction; (2) feature matching; and (3) regularization using a class of acceptable transformations that is specified through a parametric or nonparametric model. This work discusses how tools derived from

[*]Equal Contribution.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

optimal transport (OT) theory [87] can improve the second step of this pipeline (feature matching) on challenging problems. To put these results in context, we first present an overview of related methods.

1. Feature extraction. To establish spatial correspondences, one first computes descriptive local features. When dealing with (possibly annotated) point clouds, a simple choice is to rely on Cartesian coordinates (x, y, z) [3, 26]. Going further, stronger descriptors capture local geometric and topological properties: examples include shape orientation and curvatures [21, 96], shape contexts [6], spectral eigenvalues [70, 84] and annotations such as color [76] or chemical fingerprints [43, 113]. Recently, expressive feature representations have also been *learned* using deep neural networks (DNN): see [16] and subsequent works on *geometric* deep learning. Generally, feature extractors are designed to make shape registration as unambiguous as possible. In order to get closer to the ideal case of landmark matching [11], we associate discriminative features to the salient points of our shapes: this increases the robustness of the subsequent *matching* and *regularization* steps.

2. Feature matching. Once computed on both of the source and target shapes, feature vectors are put in correspondence with each other. This assignment is often encoded as an explicit mapping between the two shapes; alternatively, the vector field relating the shapes can be defined implicitly as the gradient of a geometric loss function that quantifies discrepancies between two distributions of features [35]:

a) A first major approach is to rely on **nearest neighbor** projections and the related chamfer [12] and Hausdorff distances [13], as in the Iterative Closest Point (ICP) algorithm [7]. This method can be softened through the use of a softmax (log-sum-exp) operator as in the many variants of the Coherent Point Drift (CPD) method [44, 72, 73, 81], or made robust to outliers in the specific context of rigid and affine registrations [14, 41, 128, 129].

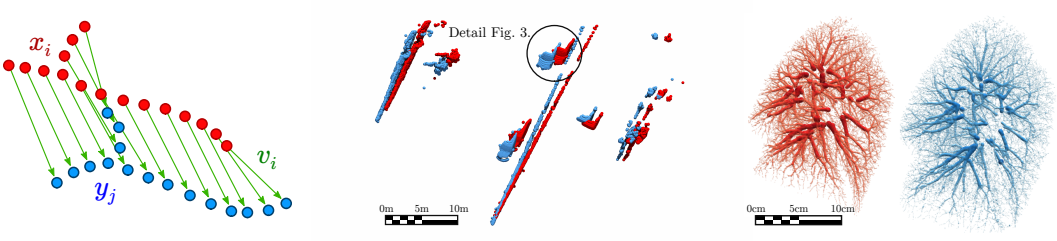
b) Alternatively, a second approach is to rely on **convolutional kernel norms** such as the Energy Distance [94], which are also known as Maximum Mean Discrepancies (MMD) in statistics [48]. These loss functions are common in imaging science [88] and computational anatomy [21, 115] but are prone to vanishing gradients [39, 40].

c) Finally, a third type of approach is to rely on **optimal transport (OT)** theory [87] and solutions of the earth mover's problem [97]. This method is equivalent to a nearest neighbor projection under a global constraint of bijectivity that enforces consistency in the matching. On the one hand, OT has been known to provide reliable correspondences in computer vision for more than two decades [26, 47, 64]. On the other hand, it has often faced major issues of scalability and robustness to outliers on noisy data. As detailed below, the main purpose of this work is to overcome these limitations and enable the widespread use of OT tools for challenging registration problems.

3. Regularization with a deformation model. The output of the two steps above is a non-smooth vector field that may not be suitable for downstream tasks due to e.g. tears and compression artifacts. As a third step, most registration methods thus rely on regularization to obtain plausible deformations. This process is task-specific, with applications that range from rigid registration [2, 30, 46, 121, 122, 133] to free-form motion estimation [49, 69, 91, 126]. In Sec. 3, we address the interaction of OT matching layers with a varied collection of regularization strategies – from optimization-based spline and diffeomorphic models to DNNs.

Recent progresses. Research works on shape registration combine ideas from the three paragraphs above to best fit the characteristics of computer vision problems [31, 71, 107]. Over the past few years, significant progress has been made on all fronts. On the one hand, (geometric) deep learning networks have been used to define data-driven feature maps [92, 92, 123] and multiscale regularization modules [68, 108, 126], sometimes fused within end-to-end architectures [30, 91, 132, 133]. On the other hand, nearest neighbor projections, kernel convolutions and transport-based matching strategies have all been generalized to take advantage of these modern descriptors: they can now be used in high-dimensional feature spaces [37, 59].

Challenges. Nevertheless, state-of-the-art (SOTA) methods in the field still have important limitations. First, modern deep learning pipelines are often hard to train to "pixel-perfect" accuracy on non-smooth shapes, with diminishing returns in terms of model size and training data [2]. Second, scaling up point neural networks to finely sampled shapes ($\mathrm{N} > 10\mathrm{k}$ points) remains a challenging research topic [37, 49, 135]. Third, the impact of the choice of a specific feature matching method on the performance of deep learning models remains only partially understood [58].



(a) Source and target points in dimension D = 2.

(b) Source and target frames, sampled with 30k points each.

(c) Exhale and inhale trees, sampled with 60k points each.

Figure 1: **Robust Optimal Transport (RobOT)** generalizes sorting to spaces of dimension $D \geqslant 1$. (a) RobOT is equivalent to a nearest neighbor projection subject to mass distribution constraints that make it robust to translations and small deformation. We demonstrate that RobOT is now ready to be part of the standard toolbox in computer vision with extensive numerical experiments for 3D scene flow estimation (b) and lung registration (c). Rendering done with Paraview [1] and PyVista [112].

Related works. Following major progress on computational OT in the mathematical literature [27, 66, 79, 103], improved modules for feature matching have attracted interest as a possible solution to these challenges. Works on sliced partial OT [9] and dustbin-OT [28] have shown that outliers can be handled effectively by OT methods for rigid registration, beyond the classic Robust Point Matching method (RPM) [26, 47]. Going further, the Sinkhorn algorithm for entropy-regularized OT [27, 64, 65, 104] has been studied extensively for shape registration in computational anatomy [36, 45] and computer graphics [33, 74, 85]. The Gromov–Wasserstein distance has also been used for shape analysis [110, 118], albeit at a higher computational cost. These applications have driven interest in the development of a complete theory for **Robust Optimal Transport** (RobOT), outlined in Sec. 2, which handles both sampling artifacts and outliers [24, 25, 67, 80, 105, 106]. Most recently, this framework has started to be used in shape analysis with applications to shape matching [36], the segmentation of brain tractograms [38] and deep deformation estimation with the FLOT architecture [91].

Contributions. We build upon the work above to tackle challenging point cloud registration problems for scene flow estimation and computational anatomy. Our **key contributions** are:

1. **Accurate feature matching with scalable OT solvers.** For the first time, we scale up RobOT for deep feature matching to high-resolution shapes with more than 10k points. To this end, we leverage the latest generation of OT solvers [35, 39] and overcome significant issues of memory usage and numerical stability. This allows us to handle fine-grained details effectively, which is key for e.g. most medical applications.
2. **Interaction with task-specific regularization strategies.** We show how to interface RobOT matchings with advanced deformation models. This is in contrast with e.g. the FLOT architecture, which focuses on the direct prediction of a vector field and cannot be used for applications that require guarantees on the smoothness of the registration.
3. **Challenging new dataset.** We release a large dataset of lung vascular trees that should be registered between inhalation and exhalation. This relevant medical problem involves large and complex deformations of high-resolution 3D point clouds. As a new benchmark for the community, we provide two strong baselines that rely respectively on global feature matching and on deep deformation estimation.
4. **Consistent SOTA performance.** Our proposed models achieve SOTA results for scene flow on Kitti [77, 78] and for point-cloud-based lung registration on DirLab-COPDGene [19]. Notably, we show that RobOT is highly suited to fine-tuning tasks: it consistently turns "good" matchings into nearly perfect registrations at an affordable numerical cost.

Main experimental observations. Is OT relevant in the deep learning era? To answer this question decisively, we perform extensive numerical experiments and ablation studies. We fully document "underwhelming" results in the Supplementary Material and distill the key lessons that we learned in the Deep-RobOT architecture (Section 3.2). This model relies on fast RobOT layers to cover for the main weaknesses of point neural networks for shape registration. It is remarkably easy to deploy and generalizes well from synthetic training data to real test samples. We thus believe that it will have a stimulating impact on both of the computer vision and medical imaging literature.

2 Robust optimal transport

This section introduces the mathematical foundations of our work. After a brief overview of Robust Optimal Transport (RobOT) theory, we discuss the main challenges that one encounters when using this framework for computer vision. To avoid memory overflows and numerical stability issues, we introduce the weighted "RobOT matching": a vector field that summarizes the information of a full transport plan with a linear memory footprint. As detailed in the next sections, this representation lets us scale up to high-resolution shapes without compromising on accuracy.

2.1 Mathematical background

The assignment problem. If $\mathrm{A} = (x_1, \dots, x_\mathrm{N})$ and $\mathrm{B} = (y_1, \dots, y_\mathrm{M})$ are two **point clouds** in $\mathbb{R}^3$ with $\mathrm{N} = \mathrm{M}$, the assignment problem between A and B reads:

$$\text{Assignment}(\mathrm{A}, \mathrm{B}) \;=\; \min_{s:[\![1,\mathrm{N}]\!]\to[\![1,\mathrm{N}]\!]} \;\frac{1}{2\mathrm{N}} \sum_{i=1}^{\mathrm{N}} \|x_i - y_{s(i)}\|_{\mathbb{R}^3}^2 \;, \quad \text{where } s \text{ is a permutation.} \tag{1}$$

This problem generalizes sorting to $\mathbb{R}^3$: if the points x_i and y_j all belong to a line, the optimal permutation s^* corresponds to a non-decreasing re-ordering of the point sets A and B [87].

Robust optimal transport. Further, OT theory allows us to consider problems where $\mathrm{N} \neq \mathrm{M}$. **Non-negative weights** $\alpha_1, \dots, \alpha_\mathrm{N}$, $\beta_1, \dots, \beta_\mathrm{M} \geqslant 0$ are attached to the points x_i, y_j and account for variations of the sampling densities, while **feature vectors** $p_1, \dots, p_\mathrm{N}$ and $q_1, \dots, q_\mathrm{M}$ in $\mathbb{R}^\mathrm{D}$ may advantageously replace raw point coordinates x_i and y_j in $\mathbb{R}^3$. Following [25, 67], the robust OT problem between the shapes $\mathrm{A} = (\alpha_i, x_i, p_i)$ and $\mathrm{B} = (\beta_j, y_j, q_j)$ reads:

$$\mathrm{OT}_{\sigma,\tau}(\mathrm{A}, \mathrm{B}) = \min_{(\pi_{i,j}) \in \mathbb{R}_{\geqslant 0}^{\mathrm{N}\times\mathrm{M}}} \;\sum_{i=1}^{\mathrm{N}} \sum_{j=1}^{\mathrm{M}} \pi_{i,j} \cdot \tfrac{1}{2}\|p_i - q_j\|_{\mathbb{R}^\mathrm{D}}^2 \tag{2}$$

$$+\; \underbrace{\sigma^2\, \mathrm{KL}\big(\pi_{i,j} \,||\, \alpha_i \otimes \beta_j\big)}_{\text{Entropic blur at scale } \sigma.} \;+\; \underbrace{\tau^2\, \mathrm{KL}\big(\textstyle\sum_j \pi_{i,j} \,||\, \alpha_i\big)}_{\pi \text{ should match A...}} \;+\; \underbrace{\tau^2\, \mathrm{KL}\big(\textstyle\sum_i \pi_{i,j} \,||\, \beta_j\big)}_{\dots \text{onto B.}} \;,$$

for any choice of the regularization parameters $\sigma > 0$ and $\tau > 0$. In the equation above, the Kullback-Leibler divergence $\mathrm{KL}\,(a_i||b_i) = \sum a_i \log(a_i/b_i) - a_i + b_i$ is a relative entropy that penalizes deviations of a non-negative vector of weights (a_i) to a reference measure (b_i).

Parameters. The first regularization term is scaled by the square of a **blur** radius σ. This characteristic length quantifies the fuzziness of the probabilistic assignment $(\pi_{i,j})$ between points x_i and y_j [35]. The last two regularization terms promote the matching of the full distribution of points A onto the target shape B: they generalize the constraints of injectivity and surjectivity of Eq. (1) to the probabilistic setting. They are scaled by the square of a maximum **reach** distance τ: this parameter acts as a soft upper bound on the distance between feature vectors p_i and q_j that should be matched with each other [38, 105].

For shape registration, we use simple heuristics for the values of these two characteristic scales: the **blur** σ should be equal to the average sampling distance in feature space $\mathbb{R}^\mathrm{D}$ while the **reach** τ should be equal to the largest plausible displacement for any given feature vector p_i. These rules are easy to follow if point features correspond to Cartesian coordinates x_i and y_j in $\mathbb{R}^3$ but may lead to unexpected behaviors if features are output by a DNN. In the latter case, we thus normalize our feature vectors so that $\|p_i\|_{\mathbb{R}^\mathrm{D}} = \|q_j\|_{\mathbb{R}^\mathrm{D}} = 1$ and pick values for σ and τ between 0 and 2.

Working with a soft, probabilistic transport plan. As detailed in [35], scalable OT solvers for Eq. (2) return a pair of dual vectors $(f_i) \in \mathbb{R}^\mathrm{N}$ and $(g_j) \in \mathbb{R}^\mathrm{M}$ that encode **implicitly** an optimal transport plan $(\pi_{i,j}) \in \mathbb{R}^{\mathrm{N}\times\mathrm{M}}$ with coefficients:

$$\pi_{i,j} \;=\; \alpha_i \beta_j \,\cdot\, \exp \tfrac{1}{\sigma^2}\big[f_i + g_j - \tfrac{1}{2}\|p_i - q_j\|_{\mathbb{R}^\mathrm{D}}^2\big] \;\geqslant\; 0\,. \tag{3}$$

In the limit case where σ tends to 0 and τ tends to $+\infty$, for generic point clouds (x_i) and (y_j) with $\mathrm{N} = \mathrm{M}$ and equal weights $\alpha_i = \beta_j = 1/\mathrm{N}$, $\pi_{i,j}$ is a permutation matrix [87]. We retrieve the simple

assignment problem of Eq. (1): $\pi_{i,j} = 1/\mathrm{N}$ if $j = s^*(i)$ and 0 otherwise. However, in general the transport plan must be understood as a probabilistic map between the point distributions A and B that assigns a weight $\pi_{i,j}$ to the coupling "$x_i \leftrightarrow y_j$". For shape registration, this implies that the main difficulties for using robust OT are two-fold: first, the coupling π is not one-to-one, but one-to-*many*; second, the lines and columns of the transport plan π do not sum up to one. Notably, this implies that when $\tau < +\infty$, the gradient of the OT cost with respect to the point positions x_i is not homogeneous: we observe vanishing and inflated values across the domain [105].

2.2 RobOT: a convenient representation of the optimal transport plan

The weighted RobOT matching. To work around these issues, we introduce the vector field:

$$v_i \;=\; \frac{\sum_{j=1}^{\mathrm{M}} \pi_{i,j} \cdot (y_j - x_i)}{\sum_{j=1}^{\mathrm{M}} \pi_{i,j}} \;\in\; \mathbb{R}^3 \qquad \text{with confidence weights} \qquad w_i \;=\; \sum_{j=1}^{\mathrm{M}} \pi_{i,j} \;\geqslant\; 0\,. \tag{4}$$

This object has the same memory footprint as the input shape A and summarizes the information that is contained in the N-by-M transport plan $(\pi_{i,j})$ – a matrix that is often too large to be stored and manipulated efficiently. This "weighted RobOT matching" is at the heart of our approach and generalizes the standard Monge map from classical OT theory [87] to the setting of (deep) shape registration. In practice, the weighted vector field $(w_1, v_1), \dots, (w_{\mathrm{N}}, v_{\mathrm{N}})$ is both convenient to use and easy to compute on GPUs. Let us briefly explain why.

Fast implementation. Our differentiable RobOT layer takes as input the two shapes $\mathrm{A} = (\alpha_i, x_i, p_i)$ and $\mathrm{B} = (\beta_j, y_j, q_j)$, with feature vectors p_i and q_j in $\mathbb{R}^{\mathrm{D}}$ that have been computed upstream using e.g. a point neural network. It returns the N vectors v_i with weights w_i that map the source points x_i onto the targets y_j in $\mathbb{R}^3$. Starting from the input point features p_i, q_j and weights α_i, β_j, we first compute the optimal dual vectors f_i and g_j using the fast solvers of the GeomLoss library [39]. We then combine Eq. (3) with Eq. (4) to compute the RobOT vectors v_i and weights w_i with $O(\mathrm{N} + \mathrm{M})$ memory footprint using the KeOps library [20, 37] for PyTorch [86] and NumPy [117]. We use a log-sum-exp formulation to ensure numerical stability. Remarkably, our implementation scales up to $\mathrm{N}, \mathrm{M} = 100\mathrm{k}$ in fractions of a second. Unlike common strategies that are based on dense or sparse representations of the transport plan π, our approach is perfectly suited to a *symbolic* implementation [37] and streams well on GPUs with optimal, contiguous memory accesses.

Comparison with nearest neighbor projections. We use our RobOT layer as a plug-in replacement for closest point matching [7]. The *blur* (σ) and *reach* (τ) scales play similar roles to the standard deviation (σ) and weight of the uniform distribution (w) in the Coherent Point Drift (CPD) method [81]: they allow us to smooth the matching in order to increase robustness to sampling artifacts.

The main difference between projection-based matching and RobOT is that the latter enforces a mass distribution constraint between the source and the target. This prevents our matching vectors from accumulating on the boundaries of the distributions of point features $(p_1, \dots, p_{\mathrm{N}})$ and $(q_1, \dots, q_{\mathrm{M}})$ in $\mathbb{R}^{\mathrm{D}}$ [40]. This property is most desirable when the shapes to register are fully observed, with a dense sampling: as detailed in Suppl. A.5, enforcing the **global consistency** of a matching is then a worthwhile registration prior.

Partial registration. On the other hand, we must also stress that OT theory has known limitations [35]. First of all, the RobOT matching cannot guarantee the preservation of remarkable points or of the shapes' topologies "on its own": it should be combined with relevant feature extractors and regularizers. Going further, partial registration is not a natural fit for standard OT formulations which assume that *all* points from both shapes must be put in correspondence with each other.

To mitigate this issue, RobOT leverages the theory of *unbalanced* optimal transport [24, 25, 67, 105, 106]: we rely on *soft* Kullback-Leibler penalties to enforce a matching between the shapes A and B in Eq. (2). In practice, the RobOT confidence weights w_i of Eq. (4) act as an attention mechanism: they vanish when no target feature vector q_j can be found in a τ-neighborhood of the source vector p_i in $\mathbb{R}^{\mathrm{D}}$, where τ is the *reach* scale that is associated to Eq. (2). This lets our registration method focus on reliable matches between similar features, without being fooled by strong constraints of bijectivity. As detailed in Suppl. A.3, combining standard FPFH features [100] with the rigid projection of Eq. (5) allows us to register partially observed shapes that have little overlap with each other.

3 Regularization and integration with a deep learning model

3.1 Smooth-RobOT: algebraic and optimization-based regularization

Notations. We now detail how to interface the weighted RobOT matching with regularization models and feature extractors that may be handcrafted [100, 101, 114] or learnt using a deep neural network [29, 30, 46, 133]. Recall that we intend to register a source point cloud $x_1, \dots, x_{\mathrm{N}}$ onto a target $y_1, \dots, y_{\mathrm{M}}$ in $\mathbb{R}^3$. Non-negative weights $\alpha_1, \dots, \alpha_{\mathrm{N}}$ and $\beta_1, \dots \beta_{\mathrm{M}} \geqslant 0$ let us take into account variations in the sampling densities and we assume that point features $p_1, \dots, p_{\mathrm{N}}$, $q_1, \dots, q_{\mathrm{M}}$ in $\mathbb{R}^{\mathrm{D}}$ have been computed upstream by a relevant feature extractor. For every source point x_i, the RobOT matching layer then provides a **desired displacement** v_i in $\mathbb{R}^3$ with **influence weight** $w_i \geqslant 0$.

Smoothing in closed form. Standard computations let us derive closed-form expressions for rigid and affine registration [3, 57, 81]. These respectively correspond to transformations:

$$\text{(Rigid RobOT)} \qquad x \in \mathbb{R}^3 \;\mapsto\; (x - x_c)\, U V^\top + x_c + v_c \in \mathbb{R}^3 \,, \tag{5}$$

$$\text{(Affine RobOT)} \qquad x \in \mathbb{R}^3 \;\mapsto\; (x - x_c)\, (\hat{X}^\top W \hat{X})^{-1} (\hat{X}^\top W \hat{Y}) + x_c + v_c \in \mathbb{R}^3 \,, \tag{6}$$

where $W = \mathrm{Diag}(w_i) \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$ is the diagonal matrix of influence weights, $x_c = \sum_i w_i x_i / \sum_i w_i \in \mathbb{R}^3$ is the barycenter of the source shape, $v_c = \sum_i w_i v_i / \sum_i w_i \in \mathbb{R}^3$ is the average desired displacement, $\hat{X} = (x_i - x_c) \in \mathbb{R}^{\mathrm{N}\times 3}$ is the centered matrix of source positions, $\hat{Y} = (x_i + v_i - x_c - v_c) \in \mathbb{R}^{\mathrm{N}\times 3}$ is the centered matrix of desired targets and $USV^\top$ is the singular value decomposition of $\hat{X}^\top W \hat{Y} \in \mathbb{R}^{3\times 3}$. This corresponds to a weighted Kabsch algorithm [60]. Likewise, we implement free-form spline registration using the KeOps library [20, 37]. A Nadaraya–Watson interpolator with kernel $k : (x, y) \in \mathbb{R}^3 \times \mathbb{R}^3 \mapsto k(x, y) > 0$ [82, 124] induces a transformation:

$$\text{(Spline RobOT)} \qquad x \in \mathbb{R}^3 \;\mapsto\; x \;+\; \textstyle\sum_{i=1}^{\mathrm{N}} w_i k(x_i, x) v_i \;/\; \sum_{i=1}^{\mathrm{N}} w_i k(x_i, x) \in \mathbb{R}^3 \,. \tag{7}$$

Black-box deformation models. Going further, we interface RobOT matchings with arbitrary deformation modules $\mathrm{Morph} : (\theta, x_i) \mapsto \hat{y}_i \in \mathbb{R}^3$ parameterized by a vector θ in $\mathbb{R}^{\mathrm{P}}$. If $\mathrm{Reg}(\theta)$ denotes a regularization penalty on the parameter θ (e.g. a squared Euclidean norm), we use standard optimizers such as L-BFGS-B [136] and Adam [63] to find the optimal deformation parameter:

$$\theta^* = \operatorname*{arg\,min}_{\theta \in \mathbb{R}^{\mathrm{P}}} \;\; \mathrm{Reg}(\theta) + \textstyle\sum_{i=1}^{\mathrm{N}} w_i \| x_i + v_i - \mathrm{Morph}(\theta, x_i) \|^2_{\mathbb{R}^3} \,. \tag{8}$$

This optimization-based method is especially relevant in the context of computational anatomy, where smooth and invertible deformations are commonly defined through the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework [5, 8, 35]. We stress that different transformation models may result in different registration results and refer to Suppl. A.3 for further details.

Optimization-based approaches provide strong geometric guarantees on the final matching. But unfortunately, these often come at a high computational price: to register complex shapes, quasi-Newton optimizers require dozens of evaluations of the deformation model $\mathrm{Morph}(\theta, x)$ and of its gradients. In practice, fitting a complex model to a pair of high-resolution shapes may thus take several minutes or seconds [17]. This precludes real-time processing and hinders research on advanced deformation models.

3.2 Deep-RobOT: registration via deep deformation prediction

Deformation prediction. In this context, there is growing interest in *fast* learning methods that avoid the use of iterative optimizers. The idea is to train a deep neural network $\mathrm{Pred} : (x_i, y_j) \mapsto \theta$ that takes as input two point clouds $\mathrm{A} = (x_1, \dots, x_{\mathrm{N}})$, $\mathrm{B} = (y_1, \dots, y_{\mathrm{M}})$ and **directly predicts the optimal vector of parameters** θ for a transformation model $\mathrm{Morph}(\theta, \cdot)$ that should map A onto B. Assuming that the prediction network Pred has been trained properly, this strategy enables real-time processing while leveraging the task-specific geometric priors that are encoded within the deformation model.

Over the last few years, numerous authors have worked in this direction for rigid registration [2, 30, 46, 121, 122, 133] and scene flow estimation [49, 69, 91, 126]. Comparable research on diffeomorphic models has focused on images that are supported on a dense grid, with successful applications to e.g. the registration of 3D brain volumes [4, 107, 131]. As of 2021, prediction-based approaches have thus become standard methods for 3D shape registration.

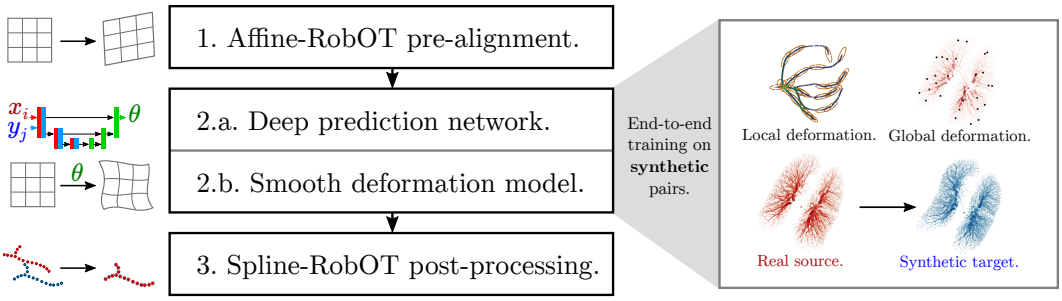


Figure 2: **The D-RobOT architecture. Left:** We apply three successive registration modules that bring the moving source shape increasingly close to the fixed target point cloud. On the one hand, the RobOT-based pre-alignment (1) and fine-tuning (3) steps take the (x, y, z) coordinates as input features and do not require any training. On the other hand, our deep registration module (2) relies on a multi-scale point neural network "Pred : $(x_i, y_j) \mapsto \theta$" (2.a) and a task-specific deformation model "Morph$(\theta, x_i) \mapsto \hat{y}_i$" (2.b). We train it end-to-end on a dataset of synthetic pairs of shapes with known ground truth correspondences. **Right:** To generate these pairs, we apply random deformations to real source shapes. For lung registration, we apply successively a vessel-preserving local perturbation and a smooth global deformation – as detailed in Suppl. A.2.

The D-RobOT model. In practice though, prediction methods still face three major challenges:

1. Common architectures may not be equivariant to **rigid or affine transformations**.
2. Dense 3D annotation is expensive, especially in a medical context. As a consequence, most predictors are trained on **synthetic data** and have to overcome a sizeable **domain gap**.
3. Predicted registrations may be **less accurate** than optimization-based solutions. Training registration networks to pixel-perfect accuracy is notoriously hard, with diminishing returns in terms of model size and number of training samples.

We propose to address these issues using RobOT layers for pre-alignment and post-processing. Our Deep RobOT (D-RobOT) model is an end-to-end deep learning architecture that is made up of three consecutive steps that we illustrate in Figure 2 and detail in Suppl. A.4:

1. **OT-based pre-alignment.** We use the rigid or affine S-RobOT models of Eqs. (5-6) to normalize the pose of the source shape. This is a fast and differentiable pre-processing.
2. **Deep registration module.** We combine a deep predictor with a task-specific deformation model to register the pre-aligned source onto the target. For the prediction network Pred : $(x_i, y_j) \mapsto \theta$, we use a multiscale point neural network that is adapted from the PointPWC-Net architecture [126]. We refer to Suppl. A.4 for a full description of our architecture and training loss.
3. **OT-based post-processing.** In order to reach "pixel-perfect" accuracy, we use the spline S-RobOT deformation model of Eq. (7) with a task-specific kernel $k(x, y)$.

Complementary strengths and weaknesses. We apply these three steps successively, which brings the moving source $\mathrm{A} = (x_1, \dots, x_\mathrm{N})$ increasingly close to the fixed target $\mathrm{B} = (y_1, \dots, y_\mathrm{M})$. Remarkably, each step of our method covers for the weaknesses of the other modules: the RobOT-based **pre-alignment** makes our pipeline robust to changes of the 3D acquisition parameters; our multiscale neural **predictor** is able to match corresponding key points quickly, even in complex situations; the domain-specific **deformation model** acts as a regularizer and improves the generalization properties of the deep registration module; the RobOT-based **fine-tuning** improves accuracy and helps our model to overcome the domain gap between synthetic and real shape data.

As detailed below, the D-RobOT model generalizes well outside of its training dataset and outperforms state-of-the-art methods on several challenging problems. We see it as a pragmatic architecture for shape registration, which is easy to deploy and tailor to domain-specific requirements. As discussed in Suppl. A.4, we found that introducing sensible geometric priors through our RobOT layers and the deformation model Morph : $(\theta, x_i) \mapsto \hat{y}_i$ results in a "forgiving" pipeline: **our model produces accurate results, even when trained on synthetic data that is not very realistic**. In a context where generating plausible 3D deformations is easier than developing custom registration models for every single task (e.g. in computational anatomy), we believe that this is an important observation.

4 Scene flow estimation

Benchmark. We now evaluate our method on a standard registration task in computer vision: the estimation of scene flow between two successive views of the same 3D scene. We follow the same experimental setting as in [49, 126], with full details provided in Suppl. A.4.3:

1. We train on the synthetic **Flying3D** dataset [75], which is made up of multiple moving objects that are sampled at random from ShapeNet. We take 19,640 pairs of point clouds for training, with dense ground truth correspondences.
2. We evaluate on 142 scene pairs from **Kitti**, a real-world dataset [77, 78]. We conduct experiments using 8,192 and 30k points per scan, sampled at random from the original data.

Performance metrics. We evaluate all methods as in [126]: **EPE3D** is the average 3D error, in centimeters; **Acc3DS** is the percentage of points with 3D error < 5 cm or relative error $< 5\%$; **Acc3DR** is the percentage of points with 3D error < 10 cm or relative error $< 10\%$; **Outliers3D** is the percentage of points with 3D error > 30 cm or relative error $> 10\%$; **EPE2D** is the average 2D error obtained by projecting the point clouds onto the image plane, measured in pixels; **Acc2D** is the percentage of points with 2D error < 3 px or relative error $< 5\%$. As detailed in Suppl. A.6, all run times were measured on a single GPU (24GB NVIDIA Quadro RTX 6000).

Methods. We study a wide range of methods and report the relevant metrics in Fig. 4:
In the **upper third** of the table, we report results for unsupervised methods that do not require ground truth correspondences for training. This includes the "raw" RobOT matching of Eq. (4), computed on (x, y, z) coordinates in $\mathbb{R}^3$ with a *blur* scale $\sigma = 1$ cm and a *reach* scale $\tau = +\infty$. Please also note that PWC refers to an improved version of PointPWC-Net, released on GitHub (`https://github.com/DylanWusee/PointPWC`) after the publication of [126] with a self-supervised loss.
In the **central third** of the table, we benchmark a collection of state-of-the-art point neural networks.
In the **lower third** of the table, we study the influence of our RobOT-based layers. The methods "Pre + FLOT/PWC + Post" correspond to the FLOT and PointPWC-Net architectures, with the additional pre-alignment and post-processing modules of Sec. 3.2. The last two lines correspond to the full D-RobOT architecture (with a spline deformation model) whose training is detailed in Suppl. A.4.3.

Results. We make three major observations:

1. Without any regularization or training, a simple RobOT matching on high-resolution data outperforms many deep learning methods in terms of speed, memory footprint and accuracy (line 6 of the table). This surprising result is strong evidence that **geometric methods and baselines deserve more attention** from the computer vision community.
2. In the lower third of the table, RobOT-enhanced methods **consistently outperform state-of-the-art methods** by a wide margin.
3. As shown in Fig. 3, these improvements are most significant on **high-resolution data**.

Overall, as detailed in Suppl. A.4.3 and A.5, we observe that optimal transport theory is especially well suited to scene flow estimation. Assuming that ground points have been removed from the 3D frames (as a standard pre-processing), most object displacements can be explained as translations and small rotations: this is an ideal setting for our robust geometric method.

5 Registration of high-resolution lung vessel trees

PVT1010: a new dataset for lung registration. Going further, we introduce a new dataset of 1,010 pairs of pulmonary vessel trees that must be registered between expiration (source) and inspiration (target). Due to the intricate geometry of the lung vasculature and the complexity of the breathing motion, the registration of these shapes is a real challenge.

As detailed in Suppl. A.1, we encode our $1{,}010 \times 2$ vessel trees as high-resolution 3D point clouds ($\mathrm{N} = \mathrm{M} = 60$k points per tree). For each point, we also provide a local estimate of the vessel radius that we use as an additional point feature or as a weight α_i or β_j in Eq. (2). Our first 1,000 pairs of 3D point clouds are provided without ground truth correspondences; for all of our experiments, we randomly sample 600 training and 100 validation cases from this large collection of unannotated patients. The last 10 cases correspond to the 10 DirLab COPDGene pairs [19]: they come with 300 expert-annotated 3D landmarks per lung pair, that we use to test our methods.

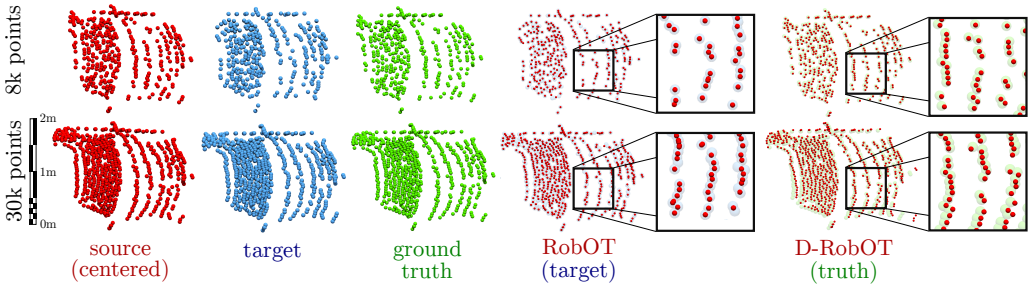


Figure 3: **Influence of the sampling density** on a detail (top car) of the Kitti frames of Fig. 1.b. **First row:** On sub-sampled 3D scenes, the regularizing priors of the D-RobOT architecture prevent over-fitting to the random sampling patterns of the target point cloud (blue). The D-RobOT output (last column, red) is very close to the ground truth scene flow (green). **Second row:** Increasing the number of points per frame reduces the influence of sampling artifacts. The simple RobOT baseline (fourth column, red) still over-fits to the target (blue) but becomes remarkably competitive.

	Method	Points	Time ms ↓	Memory Mb ↓	EPE3D cm ↓	Acc3DS % ↑	Acc3DR % ↑	Outliers3D % ↓	EPE2D px ↓	Acc2D % ↑
Unsupervised	ICP (rigid) [7]	8k	224	**2**	51.81	6.69	16.67	87.12	27.6752	10.56
	FGR (rigid) [134]	8k	—	30	48.35	13.31	28.51	77.61	18.7464	28.76
	CPD (non-rigid) [81]	8k	34,880	798	41.44	20.58	40.01	71.46	27.0583	19.80
	PWC (self) [126]	8k	237	1,016	25.49	23.79	49.57	68.63	8.9439	32.99
	RobOT (raw)	8k	170	3	**9.12**	**60.43**	**79.39**	**33.65**	**4.9920**	**56.23**
	RobOT (raw)	30k	**166**	**89**	**4.67**	**80.43**	**91.05**	**20.21**	**1.7026**	**85.71**
Supervised training on FlyingThings3D	FlowNet3D [69]	8k	—	690	17.67	37.38	66.77	52.71	7.2141	50.92
	SPLATFlowNet [111]	8k	—	—	19.88	21.74	53.91	65.75	8.2306	41.89
	original BCL [49]	8k	—	—	17.29	25.16	60.11	62.15	7.3476	44.11
	HPLFlowNet [49]	8k	—	—	11.69	47.83	77.76	41.03	4.8055	59.38
	FLOT [49]	8k	324	2,826	5.51	75.79	90.98	23.95	3.3152	75.10
	PWC [126]	8k	**237**	1,016	5.28	85.83	94.08	18.85	3.0074	81.48
	PWC	30k	1,138	10,691	7.63	67.54	92.30	26.09	3.5212	70.74
	Pre + FLOT + Post	8k	487	2,826	5.33	76.84	91.65	23.56	3.2786	75.31
	Pre + PWC + Post	8k	279	1,034	3.35	90.10	97.32	**16.20**	**1.4301**	**93.85**
	Pre + PWC + Post	30k	1,207	10,691	3.50	90.04	96.71	17.28	1.5917	90.37
	D-RobOT (spline)	8k	268	**396**	**3.15**	**90.51**	**97.42**	16.26	1.4532	93.76
	D-RobOT (spline)	30k	**547**	**610**	**2.23**	**95.88**	**99.19**	**12.89**	**1.0336**	**96.75**

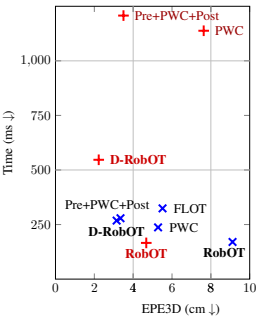
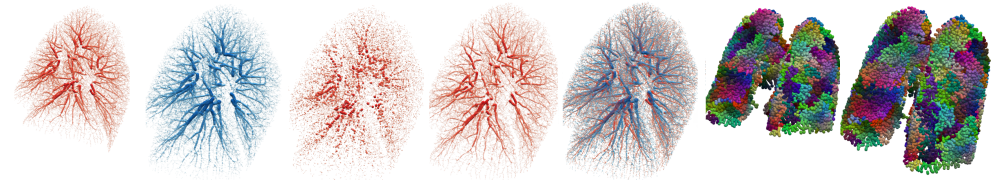
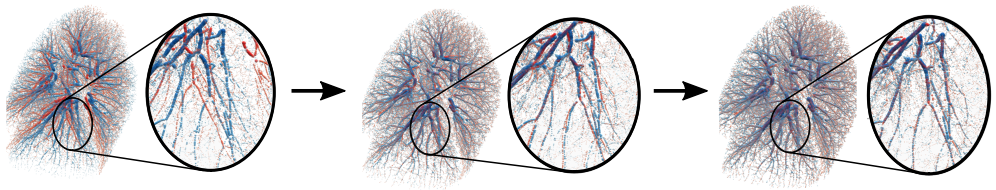


Figure 4: **Evaluation on the Kitti dataset for 3D scene flow.** Black numbers and crosses (×) correspond to results on scene pairs that are sampled with 8,192 points per frame; red numbers and plus signs (+) correspond to scene pairs that are sampled with 30,000 points per frame.



(a) Source. (b) Target. (c) RobOT. (d) S-RobOT and target. (e) Deep features (source, target).



(f) After pre-alignment, target. (g) After deep registration, target. (h) After fine-tuning, target.

Figure 5: **Registration of lung vascular trees. First row: S-RobOT** registration with the deep features of Suppl. A.3.3. We display: (a) the source and (b) target shapes; (c) the "raw" RobOT registration; (d) a smoother S-RobOT registration with spline regularization, an overlap of this result with the target shape; (e) a visualization of the deep features on a pair of lung vascular trees, with colors that correspond to a t-SNE embedding of the point features in color space [116]. **Second row: D-RobOT** registration. We display the successive steps of our model: (f) pre-alignment with affine S-RobOT; (g) deep registration with an LDDMM model; (h) fine-tuning with spline S-RobOT.

Table 1: **3D registration errors on the 3,000 expert-annotated DirLab landmarks.** The † symbol denotes methods that are based on *image keypoints* [50, 51] and are evaluated on the original DirLab *image* dataset; all other approaches are tested on our *point clouds*. Due to its large memory requirements, CPD is tested on clouds of 20k points (instead of 60k).

	Method	Average error mm ↓	Percentiles 25% mm ↓	50% mm ↓	75% mm ↓	Time s ↓
No training	Input data	23.30	13.18	22.22	31.65	—
	ICP (affine) [7]	15.05	9.60	14.06	20.01	0.52
	CPD (non-rigid) [81]	9.30	5.95	8.60	11.83	332.60
	RobOT (affine)	10.45	6.01	9.83	13.97	0.18
	RobOT (raw)	9.41	4.89	8.35	13.04	**0.15**
Supervised	DGCNN-CPD† [50]	4.30	—	—	—	—
	DispEmd† [51]	3.42	—	—	—	—
	S-RobOT (spline)	5.72	3.19	5.04	7.35	2.77
	S-RobOT (LDDMM)	5.48	2.86	4.44	7.14	42.30
	D-RobOT (raw)	3.40	1.40	2.58	3.69	1.26
	D-RobOT (spline)	2.95	1.30	2.50	3.19	1.87
	D-RobOT (LDDMM)	**2.86**	**1.25**	**2.23**	**3.11**	1.92

Data augmentation. The imbalance between our large training set and the small collection of 10 test cases reveals a fundamental challenge in computational anatomy: annotating pairs of 3D medical shapes with dense correspondences is prohibitively expensive. To work around this problem, we train our networks on synthetic deformations of the 600×2 lung shapes that make up our training set. As detailed in Suppl. A.2, we use a two-scales random field to generate a wide variety of deformations. This allows us to create a suitable training set with dense "ground truth" correspondences.

Overall, as discussed in Suppl. A.4.2, we found that supervised training on synthetic deformations is both easier and more efficient than unsupervised training on real lung pairs. From the chamfer and Wasserstein distances [35] to local Laplacian penalties [126], none of the unsupervised loss functions that we experimented with was able to deal with the complex geometry of our lung vascular trees.

Methods. We benchmark a wide range of methods in Fig. 5 and Tab. 1. In the **upper half** of the table, we evaluate geometric approaches that require no training; in the **lower half** of the table, we benchmark scalable point neural networks that we trained on our synthetic dataset. We evaluate three types of RobOT-based approaches: a simple RobOT matching computed using (x, y, z) coordinates as point features, that may either be "raw" as in Eq. (4) or regularized using Eq. (6); an S-RobOT matching that we compute using the deep features of Suppl. A.3.3 and regularize with the spline smoothing of Eq. (7) or the LDDMM optimization of Eq. (10); a D-RobOT architecture that we pair with three deformation models $\text{Morph}(\theta, x_i) \mapsto \hat{y}_i$ and describe in depth in Suppl. A.4.2.

Results. We provide additional experiments in Suppl. A.4.2 and make three major observations:

1. **Explicit regularization** with a spline or LDDMM deformation model is key. Model-free architectures that predict raw 3D correspondences produce non-smooth results that are not anatomically plausible, even when they are trained entirely on smooth deformations.
2. The D-RobOT architecture combines a **high acccuracy** with fast run times.
3. Most remaining errors occur at the **boundary of the lungs**, where acquisition artifacts prevent the thinnest vessels from being sampled reliably in our point cloud representation.

6 Conclusion, limitations and future work

Our work builds upon a decade of active research in the field of computational optimal transport. We leverage major advances on RobOT solvers to define a new matching layer which is a plug-and-play replacement for nearest neighbor projection. This operation has two major uses in 3D shape registration: first, it provides a **very strong geometric baseline** for e.g. scene flow estimation; second, it increases the accuracy and generalization abilities of point neural networks on finely sampled 3D shapes. We see D-RobOT as a **mature and versatile architecture** for shape registration which is easy to train and adapt to task-specific requirements in e.g. medical imaging.

Going forward, we see three main ways of improving this work. First, we still have to investigate in depth the important problem of occlusions and **partial acquisitions**. Second, integrating **task-specific features** beyond the (x, y, z) point coordinates is often key to perfect results. In the specific setting of lung registration, working with image-based features or focusing on branching points could be a way of improving performance at the cost of portability: recent works such as [52, 53] are an excellent source of inspiration. Finally, we believe that high-quality **software packaging** is an important part of research in our field. We intend to keep working on the topic and distribute our methods through a user-friendly Python library for widespread use by the scientific community.

Acknowledgments and Disclosure of Funding

Research reported in this publication was supported by the National Heart, Lung, and Blood Institute of the National Institutes of Health under award numbers R01HL149877 and R01HL116473. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The authors would also like to thank the anonymous reviewers for their most valuable advice.

References

[1] J. Ahrens, B. Geveci, and C. Law. Paraview: An end-user tool for large data visualization. *The visualization handbook*, 717(8), 2005.

[2] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey. PointNetLK: Robust & efficient point cloud registration using PointNet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2019.

[3] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on pattern analysis and machine intelligence*, 5:698–700, 1987.

[4] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca. Voxelmorph: a learning framework for deformable medical image registration. *IEEE transactions on medical imaging*, 38(8):1788–1800, 2019.

[5] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61(2):139–157, 2005.

[6] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. *Advances in neural information processing systems*, 13:831–837, 2000.

[7] P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[8] A. Bône, M. Louis, B. Martin, and S. Durrleman. Deformetrica 4: an open-source software for statistical shape analysis. In *International Workshop on Shape in Medical Imaging*, pages 3–13. Springer, 2018.

[9] N. Bonneel and D. Coeurjolly. SPOT: sliced partial optimal transport. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019.

[10] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989.

[11] F. L. Bookstein. *Morphometric tools for landmark data: geometry and biology*. Cambridge University Press, 1997.

[12] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer vision, graphics, and image processing*, 27(3):321–345, 1984.

[13] S. Bouaziz, A. Tagliasacchi, H. Li, and M. Pauly. Modern techniques and applications for real-time non-rigid registration. In *SIGGRAPH ASIA 2016 Courses*, pages 1–25. ACM Digital Library, 2016.

[14] S. Bouaziz, A. Tagliasacchi, and M. Pauly. Sparse iterative closest point. In *Computer graphics forum*, volume 32, pages 113–123. Wiley Online Library, 2013.

[15] Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

[16] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[17] M. Brunn, N. Himthani, G. Biros, M. Mehl, and A. Mang. Fast GPU 3D diffeomorphic image registration. *Journal of Parallel and Distributed Computing*, 149:149–162, 2021.

[18] E. Castillo, R. Castillo, D. Fuentes, and T. Guerrero. Computing global minimizers to a constrained B-spline image registration problem from optimal L1 perturbations to block match data. *Medical physics*, 41(4):041904, 2014.

[19] R. Castillo, E. Castillo, D. Fuentes, M. Ahmad, A. M. Wood, M. S. Ludwig, and T. Guerrero. A reference dataset for deformable image registration spatial accuracy evaluation using the COPDgene study archive. *Physics in Medicine & Biology*, 58(9):2861, 2013.

[20] B. Charlier, J. Feydy, J. Glaunès, F.-D. Collin, and G. Durif. Kernel operations on the GPU, with autodiff, without memory overflows. *Journal of Machine Learning Research*, 22(74):1–6, 2021.

[21] N. Charon and A. Trouvé. The varifold representation of nonoriented shapes for diffeomorphic registration. *SIAM Journal on Imaging Sciences*, 6(4):2547–2580, 2013.

[22] R. T. Q. Chen, B. Amos, and M. Nickel. Learning neural event functions for ordinary differential equations. *International Conference on Learning Representations*, 2021.

[23] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 2018.

[24] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018.

[25] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard. Unbalanced optimal transport: Dynamic and Kantorovich formulations. *Journal of Functional Analysis*, 274(11):3090–3123, 2018.

[26] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.

[27] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

[28] Z. Dang, F. Wang, and M. Salzmann. Learning 3D-3D correspondences for one-shot partial-to-partial registration. *arXiv preprint arXiv:2006.04523*, 2020.

[29] H. Deng, T. Birdal, and S. Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3D local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018.

[30] H. Deng, T. Birdal, and S. Ilic. Ppfnet: Global context aware local features for robust 3D point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 195–205, 2018.

[31] N. Donati, A. Sharma, and M. Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8592–8601, 2020.

[32] J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

[33] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *arXiv preprint arXiv:2010.15261*, 2020.

[34] R. S. J. Estépar, J. C. Ross, K. Russian, T. Schultz, G. R. Washko, and G. L. Kindlmann. Computational vascular morphometry for the assessment of pulmonary vascular disease based on scale-space particles. In *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1479–1482. IEEE, 2012.

[35] J. Feydy. *Geometric data analysis, beyond convolutions*. PhD thesis, Université Paris–Saclay, 2020.

[36] J. Feydy, B. Charlier, F.-X. Vialard, and G. Peyré. Optimal transport for diffeomorphic registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 291–299. Springer, 2017.

[37] J. Feydy, J. Glaunès, B. Charlier, and M. Bronstein. Fast geometric learning with symbolic matrices. *Proc. NeurIPS*, 2(4):6, 2020.

[38] J. Feydy, P. Roussillon, A. Trouvé, and P. Gori. Fast and scalable optimal transport for brain tractograms. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 636–644. Springer, 2019.

[39] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouvé, and G. Peyré. Interpolating between optimal transport and MMD using Sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690. PMLR, 2019.

[40] J. Feydy and A. Trouvé. Global divergences between measures: from Hausdorff distance to optimal transport. In *International Workshop on Shape in Medical Imaging*, pages 102–115. Springer, 2018.

[41] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[42] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *International conference on medical image computing and computer-assisted intervention*, pages 130–137. Springer, 1998.

[43] P. Gainza, F. Sverrisson, F. Monti, E. Rodola, D. Boscaini, M. Bronstein, and B. Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

[44] W. Gao and R. Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using Gaussian filter and twist parameterization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11095–11104, 2019.

[45] S. Gerber, M. Niethammer, M. Styner, and S. Aylward. Exploratory population analysis with unbalanced optimal transport. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 464–472. Springer, 2018.

[46] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser. The perfect match: 3D point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019.

[47] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2D and 3D point matching: pose estimation and correspondence. *Pattern recognition*, 31(8):1019–1031, 1998.

[48] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[49] X. Gu, Y. Wang, C. Wu, Y. J. Lee, and P. Wang. HPLFlowNet: Hierarchical permutohedral lattice FlowNet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3254–3263, 2019.

[50] L. Hansen, D. Dittmer, and M. P. Heinrich. Learning deformable point set registration with regularized dynamic graph CNNs for large lung motion in COPD patients. In *International Workshop on Graph Learning in Medical Imaging*, pages 53–61. Springer, 2019.

[51] L. Hansen and M. P. Heinrich. Tackling the problem of large deformations in deep learning based medical image registration using displacement embeddings. *Medical Imaging in Deep Learning, arXiv preprint arXiv:2005.13338*, 2020.

[52] L. Hansen and M. P. Heinrich. Deep learning based geometric registration for medical images: How accurate can we get without visual features? In *International Conference on Information Processing in Medical Imaging*, pages 18–30. Springer, 2021.

[53] L. Hansen and M. P. Heinrich. GraphRegNet: Deep graph regularisation networks on sparse keypoints for dense registration of 3d lung CTs. *IEEE Transactions on Medical Imaging*, 40(9):2246–2257, 2021.

[54] G. L. Hart, C. Zach, and M. Niethammer. An optimal control approach for deformable registration. In *CVPR Workshops*, pages 9–16, 2009.

[55] M. P. Heinrich, H. Handels, and I. J. Simpson. Estimating large lung motion in COPD patients by symmetric regularised correspondence fields. In *International conference on medical image computing and computer-assisted intervention*, pages 338–345. Springer, 2015.

[56] S. Hermann. Evaluation of scan-line optimization for 3D medical image registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3073–3080, 2014.

[57] B. K. Horn. Closed-form solution of absolute orientation using unit quaternions. *Josa a*, 4(4):629–642, 1987.

[58] X. Huang, G. Mei, J. Zhang, and R. Abbas. A comprehensive survey on point cloud registration. *arXiv preprint arXiv:2103.02690*, 2021.

[59] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.

[60] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[61] G. Kindlmann et al. Teem. `http://teem.sourceforge.net/build.html`.

[62] G. L. Kindlmann, R. S. J. Estépar, S. M. Smith, and C.-F. Westin. Sampling and visualizing creases with scale-space particles. *IEEE transactions on visualization and computer graphics*, 15(6):1415–1424, 2009.

[63] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[64] J. J. Kosowsky and A. L. Yuille. The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural networks*, 7(3):477–490, 1994.

[65] C. Léonard. From the Schrödinger problem to the Monge–Kantorovich problem. *Journal of Functional Analysis*, 262:1879–1920, 2012.

[66] B. Lévy. A numerical algorithm for L2 semi-discrete optimal transport in 3D. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1693–1715, 2015.

[67] M. Liero, A. Mielke, and G. Savaré. Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117, 2018.

[68] O. Litany, T. Remez, E. Rodola, A. Bronstein, and M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 5659–5667, 2017.

[69] X. Liu, C. R. Qi, and L. J. Guibas. FlowNet3D: Learning scene flow in 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019.

[70] H. Lombaert, L. Grady, X. Pennec, N. Ayache, and F. Cheriet. Spectral log-demons: diffeomorphic image registration with very large deformations. *International journal of computer vision*, 107(3):254–271, 2014.

[71] M. Lüthi, T. Gerig, C. Jud, and T. Vetter. Gaussian process morphable models. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1860–1873, 2017.

[72] J. Ma, J. Zhao, Y. Ma, and J. Tian. Non-rigid visible and infrared face registration via regularized Gaussian fields criterion. *Pattern Recognition*, 48(3):772–784, 2015.

[73] J. Ma, J. Zhao, and A. L. Yuille. Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on image Processing*, 25(1):53–64, 2015.

[74] M. Mandad, D. Cohen-Steiner, L. Kobbelt, P. Alliez, and M. Desbrun. Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017.

[75] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[76] H. Men, B. Gebre, and K. Pochiraju. Color point cloud registration with 4D ICP algorithm. In *2011 IEEE International Conference on Robotics and Automation*, pages 1511–1516. IEEE, 2011.

[77] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015.

[78] M. Menze, C. Heipke, and A. Geiger. Joint 3D estimation of vehicles and scene flow. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 2:427, 2015.

[79] Q. Mérigot. A multiscale approach to optimal transport. In *Computer Graphics Forum*, volume 30, pages 1583–1592. Wiley Online Library, 2011.

[80] D. Mukherjee, A. Guha, J. M. Solomon, Y. Sun, and M. Yurochkin. Outlier-robust optimal transport. In *International Conference on Machine Learning*, pages 7850–7860. PMLR, 2021.

[81] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.

[82] E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

[83] P. Nardelli, J. C. Ross, and R. S. J. Estépar. Generative-based airway and vessel morphology quantification on chest CT images. *Medical image analysis*, 63:101691, 2020.

[84] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012.

[85] G. Pai, J. Ren, S. Melzi, P. Wonka, and M. Ovsjanikov. Fast Sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *CVPR*, 2021.

[86] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. *OpenReview.net*, 2017.

[87] G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[88] J. P. Pluim, J. A. Maintz, and M. A. Viergever. Mutual information matching in multiresolution contexts. *Image and Vision Computing*, 19(1-2):45–52, 2001.

[89] T. Polzin, M. Niethammer, M. P. Heinrich, H. Handels, and J. Modersitzki. Memory efficient LDDMM for lung CT. In *International conference on medical image computing and computer-assisted intervention*, pages 28–36. Springer, 2016.

[90] T. Polzin, J. Rühaak, R. Werner, J. Strehlow, S. Heldmann, H. Handels, and J. Modersitzki. Combining automatic landmark detection and variational methods for lung CT registration. In *Fifth international workshop on pulmonary image analysis*, pages 85–96, 2013.

[91] G. Puy, A. Boulch, and R. Marlet. FLOT: Scene flow on point clouds guided by optimal transport. *arXiv preprint arXiv:2007.11142*, 2020.

[92] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[93] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[94] M. L. Rizzo and G. J. Székely. Energy distance. *Wiley interdisciplinary reviews: Computational statistics*, 8(1):27–38, 2016.

[95] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[96] P. Roussillon and J. A. Glaunès. Kernel metrics on normal cycles and application to curve matching. *SIAM Journal on Imaging Sciences*, 9(4):1991–2038, 2016.

[97] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[98] J. Rühaak, S. Heldmann, T. Kipshagen, and B. Fischer. Highly accurate fast lung CT registration. In *Medical Imaging 2013: Image Processing*, volume 8669, page 86690Y. International Society for Optics and Photonics, 2013.

[99] J. Rühaak, T. Polzin, S. Heldmann, I. J. Simpson, H. Handels, J. Modersitzki, and M. P. Heinrich. Estimation of large motion in lung CT by integrating regularized keypoint correspondences into dense deformable registration. *IEEE transactions on medical imaging*, 36(8):1746–1757, 2017.

[100] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[101] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ international conference on intelligent robots and systems*, pages 3384–3391. IEEE, 2008.

[102] R. San José Estepar et al. Chest imaging platform. `https://chestimagingplatform.org/`.

[103] B. Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019.

[104] E. Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pages 269–310, 1932.

[105] T. Séjourné, J. Feydy, F.-X. Vialard, A. Trouvé, and G. Peyré. Sinkhorn divergences for unbalanced optimal transport. *arXiv preprint arXiv:1910.12958*, 2019.

[106] T. Séjourné, F.-X. Vialard, and G. Peyré. The unbalanced Gromov Wasserstein distance: Conic formulation and relaxation. *arXiv preprint arXiv:2009.04266*, 2020.

[107] Z. Shen, X. Han, Z. Xu, and M. Niethammer. Networks for joint affine and non-parametric image registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4224–4233, 2019.

[108] Z. Shen, F.-X. Vialard, and M. Niethammer. Region-specific diffeomorphic metric mapping. *Advances in Neural Information Processing Systems*, 32:1098–1108, 2019.

[109] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524, 1968.

[110] J. Solomon, G. Peyré, V. G. Kim, and S. Sra. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG)*, 35(4):1–13, 2016.

[111] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz. SplatNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2530–2539, 2018.

[112] C. B. Sullivan and A. A. Kaszynski. PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (vtk). *Journal of Open Source Software*, 4(37):1450, 2019.

[113] F. Sverrisson, J. Feydy, B. E. Correia, and M. M. Bronstein. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15272–15281, 2021.

[114] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010.

[115] M. Vaillant and J. Glaunes. Surface matching via currents. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 381–392. Springer, 2005.

[116] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

[117] S. Van Der Walt, S. C. Colbert, and G. Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in science & engineering*, 13(2):22–30, 2011.

[118] T. Vayer, R. Flamary, N. Courty, R. Tavenard, and L. Chapel. Sliced Gromov–Wasserstein. *Advances in Neural Information Processing Systems*, 32:14753–14763, 2019.

[119] V. Vishnevskiy, T. Gass, G. Szekely, C. Tanner, and O. Goksel. Isotropic total variation regularization of displacements in parametric image registration. *IEEE transactions on medical imaging*, 36(2):385–395, 2016.

[120] V. Vishnevskiy, T. Gass, G. Szekely, C. Tanner, and O. Goksel. Isotropic total variation regularization of displacements in parametric image registration. *IEEE transactions on medical imaging*, 36(2):385–395, 2017.

[121] Y. Wang and J. M. Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019.

[122] Y. Wang and J. M. Solomon. PRNet: Self-supervised learning for partial-to-partial registration. *arXiv preprint arXiv:1910.12240*, 2019.

[123] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph CNN for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[124] G. S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372, 1964.

[125] W. Wu, Z. Qi, and L. Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

[126] W. Wu, Z. Wang, Z. Li, W. Liu, and L. 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.

[127] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[128] H. Yang and L. Carlone. A polynomial-time solution for robust registration with extreme outlier rates. *arXiv preprint arXiv:1903.08588*, 2019.

[129] H. Yang, J. Shi, and L. Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 2020.

[130] J. Yang, H. Li, D. Campbell, and Y. Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2015.

[131] X. Yang, R. Kwitt, M. Styner, and M. Niethammer. QuickSilver: Fast predictive image registration–a deep learning approach. *NeuroImage*, 158:378–396, 2017.

[132] Z. J. Yew and G. H. Lee. RPM-Net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11824–11833, 2020.

[133] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3DMatch: Learning local geometric descriptors from RGB-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017.

[134] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *European Conference on Computer Vision*, pages 766–782. Springer, 2016.

[135] Y. Zhou, C. Wu, Z. Li, C. Cao, Y. Ye, J. Saragih, H. Li, and Y. Sheikh. Fully convolutional mesh autoencoder using efficient spatially varying kernels. *arXiv preprint arXiv:2006.04325*, 2020.

[136] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997.

A Supplementary Material

We provide additional information on our RobOT layer and the two derived methods:

- **S-RobOT**: feature matching with RobOT, followed by a **smoothing** that is implemented in closed form or relies on an optimization loop – as detailed in Sec. 3.1.
- **D-RobOT**: S-RobOT as a pre- and post-processing tool for a **deep** deformation estimator – as detailed in Sec. 3.2.

More specifically:

1. Sec. A.1 provides details on **PVT1010**, our dataset of pulmonary vascular trees.
2. Sec. A.2 describes our **synthetic deformations** for augmenting PVT1010.
3. Sec. A.3 contains more information on global feature matching with **S-RobOT**.
4. Sec. A.4 provides details on our deep deformation prediction approach **D-RobOT**, with additional results on the PVT1010 and Kitti datasets.
5. Sec. A.5 discusses the differences between **RobOT and nearest neighbor projection**.
6. Sec. A.6 details our **computational resources**.
7. Finally, Sec. A.7 discusses the **societal impact** of our work.

A.1 Pulmonary vascular tree dataset

Introducing the PVT1010 dataset. We introduce a new pulmonary vascular tree dataset for point cloud registration. The PVT1010 dataset includes 1,010 pairs of inhale/exhale lung vascular trees extracted from 3D computed tomography (CT) images; 10 of these correspond to the 10 cases of the public DirLab-COPDGene [19] dataset which includes, for each pair, 300 expert annotated landmarks that are in correspondence with each other and that we use to validate our results. We extracted the lung vascular geometry in both inspiratory and expiratory CT scans using a scale-space particle system [34, 62] that is implemented in the Teem library [61]. We used the pipeline that is defined in the chest imaging platform [83, 102].

Legal and regulatory information. The vascular tree reconstructions that are used in this study were part of the COPDGene study (NCT00608764). This study has been IRB approved and participants have provided their consent. The investigators from the Brigham and Women's Hospital (Harvard Medical School) only had access to de-identified CT images to perform vascular reconstructions. Since we are performing secondary analysis using de-identified data, the work under consideration is not considered human subjects research and did not imply additional risks to the participants. Risks related to ionizing radiation exposure were described in the primary IRB-approved study. Study identifiers were re-coded for our release of PVT1010 to preserve anonymity. PVT1010 is released under the *Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License*.

Point cloud representation. We now detail how we extracted the lung vascular trees as high-resolution 3D point clouds from the raw CT images. To perform this geometric segmentation task, we rely on a system of 4D particles that are defined by three spatial coordinates plus one scale parameter that corresponds to the local radius of the lung vessel – these radii are used by RobOT as point weights α_i and β_j in Eq. (2). Our segmentation method starts from a point cloud that is initialized using the Frangi filter [42]. Then, we fit this point cloud representation to our 3D CT volumes by minimizing iteratively a system energy that is expressed as the sum of:

- An **inter-particle regularization** energy that ensures convenient sampling properties. We rely on a sum of quartic polynomials of the pairwise point distances, with a tunable potential well that is chosen to induce regular sampling at a fixed distance between the points.
- A **particle-image data fidelity** term, which is computed using the image Hessian at the current particles' locations.

As a final result, we obtain points that are approximately equally distributed along the vessel centerlines. The vessel radii are first approximated as the image scales at which the middle eigenvalues of the Hessian are locally minimized, and then refined using the generative approach of [83].

The resulting dataset has the following properties, which result in a challenging registration task:

1. The vascular trees have a **complex structure** and exhibit **complex, large motions** between inhalation and exhalation.
2. To capture the complex anatomical structure of the lungs at millimeter scale, registration methods need to focus on branching points or rely on **high-resolution** point clouds.
3. Due to the fixed image resolution of the raw CT volume and to acquisition differences between the inhale and exhale scans, the extracted inhale and exhale vascular trees are **not fully consistent** with each other. This is especially true for tiny structures at the lung boundary, that may not be visible on the smaller lung images at exhalation time.

Sampling density, resolution. The original CT images from which we extract our point clouds are acquired with a uniform resolution on the x, y and z axes: depending on the patients, the side length of our voxels varies between 0.60 mm and 0.65 mm. Using the processing above, we turn these volumetric images into 3D point clouds with 60k samples per lung vascular tree: depending on the subject, the average sampling distance (from each point to its nearest neighbor in the 3D point cloud) ranges between 0.6 mm and 1.0 mm.

We note that for our 10 test cases, the Dirlab annotations were performed on a down-sampled volume with a resolution of 2.5 mm on the z axis. To guarantee a fair comparison between image-based and point-based methods, we follow standard practice for this dataset (`https://www.dir-lab.com/Results.html`) and report our results in Tab. 1 with a "snap-to-voxel" post-processing: we quantize our 3D lung registrations on the original grids (with spacing $\sim 0.625\,\text{mm} \times 0.625\,\text{mm} \times 2.5\,\text{mm}$) before computing the average errors and percentiles. In practice, we note that this quantization slightly lowers the 25% percentile of the registration errors (as we "snap" many displacements to the correct voxel or slice) but increases the 50% and 75% percentiles (some landmarks get "snapped" to the wrong slice). Please note that in the Supplementary Material, we do not include this quantization step for e.g. ablation studies: this allows us to study more precisely the impact of each layer in our architecture.

Volume vs point cloud representation. We stress that our point clouds contain much less information than the original 3D volumes from which they have been sampled. We discard all the intensity (grayscale) values and only retain the sparse geometric support of the lung vascular tree – 60k points out of 100M+ voxels. As a consequence, our registration task on 3D point clouds is significantly harder than the original DirLab benchmark (`https://www.dir-lab.com/Results.html`). Whereas optimization-based methods on the full CT **volumes** reach a nearly perfect accuracy of 0.60 mm to 1.00 mm with run times on the order of the **minute** [18, 19, 55, 56, 89, 90, 98, 99, 119, 120], our **point** neural networks reach an average accuracy of 2 mm to 4 mm in one or two **seconds**.

The main purpose of our work on the PVT1010 dataset is to show that fast and accurate registration is now at hand, even on very degraded anatomical data. This is of significant interest for clinical practice: our method is suitable for **real-time processing** and has **built-in robustness** to changes of the CT acquisition parameters that may affect the intensities of the raw image volumes. Going forward, as detailed in the conclusion of our manuscript, we intend to work on improving the accuracy of our method with a better sampling strategy and image-based features. Packaging our method as an accessible Python toolbox will also open the door to a genuine multi-center evaluation of our trained models.

A.2 Augmentation of the training dataset for vascular tree registration

We now describe how to augment the PVT1010 dataset with synthetic deformations in order to create a large training set with **dense ground truth annotations** for lung registration. We proceed in four steps: voxel-grid sampling; local deformation; global deformation; local property distortion and degradation using an inconsistent sub-sampling. Our efficient implementation lets us generate synthetic training pairs online – just like a standard data augmentation layer. We showcase our local and global deformations in Fig. 6.

1. Voxel-grid sampling. To start, we use a voxel-grid strategy to re-sample the raw point clouds with a standard sampling density. First, we subdivide the volume space into coarse 3D blocks with spacing $s_{\text{voxelgrid}} = 0.03\,\text{mm}$. Second, we sort all points into these cubic bins according to their (x, y, z) coordinates. Third, we compute one barycenter per cubic cell to down-sample the original point cloud. Since we work with *weighted* point clouds, these local centroids are associated to the sums of the weights of all points in the corresponding cells.

2. Local deformation. We then sample control points from the vessel trees and generate random displacements that we smooth using an anisotropic spline model:

1. We first sample $\mathrm{C} = 1,000$ spline control points x_c uniformly at random from the point cloud $(x_1, \ldots, x_\mathrm{N})$.
2. Second, we compute local covariance matrices for the distribution of points x_i in a neighborhood of each control point x_c, using an isotropic Gaussian kernel window.
3. We compute the three eigenvalues (e_c^1, e_c^2, e_c^3) and unit eigenvectors (v_c^1, v_c^2, v_c^3) of each local covariance matrix to determine the main direction of the lung vessel. To avoid rank deficiency, we use a lower threshold of 0.2 on the eigenvalues e_c^k. For each control point x_c, we then normalize the vector of three eigenvalues (e_c^1, e_c^2, e_c^3) as $(e_c^1, e_c^2, e_c^3)/\sqrt{(e_c^1)^2 + (e_c^2)^2 + (e_c^3)^2}$.
4. For each control point x_c, we create a numerically stable anisotropic covariance matrix $\Sigma_c = \sum_{k=1}^3 (s_\text{local} e_c^k)^2 \, v_c^k {v_c^k}^\top$, where $s_\text{local} = 4\,\text{mm}$ is a positive scaling factor.
5. To obtain a robust estimation of the local covariance structure of our point cloud, we re-run steps 2-4 with neighborhoods that are defined using an *anisotropic* Gaussian kernel window of covariance Σ_c.
6. For every control point x_c, we generate a random displacement vector $\Delta x_c \in \mathbb{R}^3$ such that $\|\Delta x_c\| \leqslant d_\text{local}$, drawn uniformly in the ball of center 0 and radius $d_\text{local} = 3\,\text{mm}$.
7. We smooth and interpolate the displacement vector field Δx_c from the control points x_c to the full point cloud $\{x_i\}$ using an anisotropic Nadaraya–Watson kernel interpolator:

$$x_i \;\leftarrow\; x_i \;+\; \frac{\sum_{c=1}^\mathrm{C} k_{\Sigma_c}(x_c, x_i) \Delta x_c}{\sum_{c=1}^\mathrm{C} k_{\Sigma_c}(x_c, x_i)} \;, \tag{9}$$

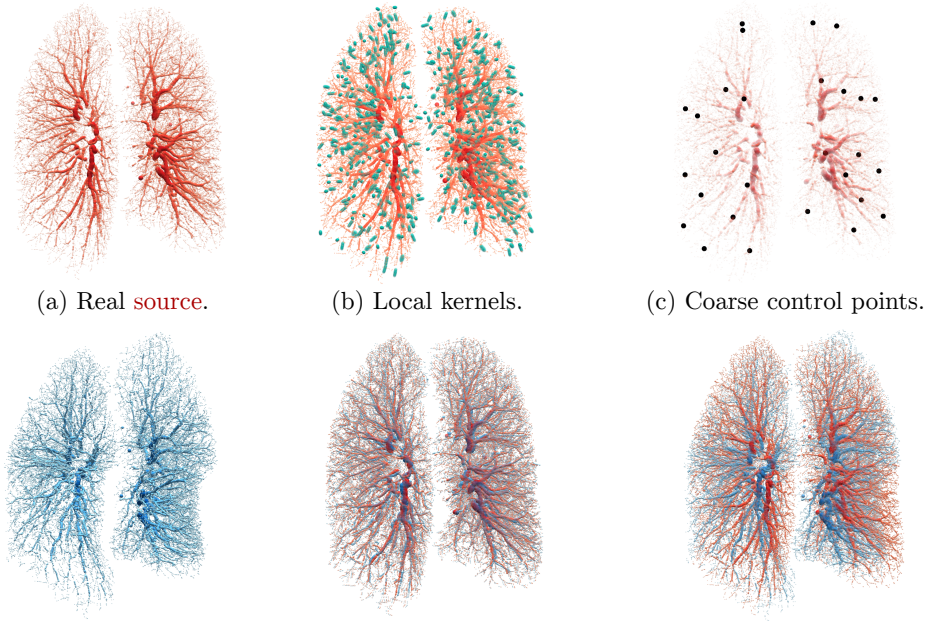
   where $k_{\Sigma_c}(x_c, \cdot)$ is a Gaussian kernel with local covariance Σ_c. This anisotropic formula ensures that the **local connectivity structure** of the lung vessel tree is preserved: the displacement of points that belong to the same vessel are strongly correlated with each other.

3. Global deformation. The step above simulates local relative displacements between lung vessels. To take large-scale breathing movements into account, we apply a second spline deformation which is smoother but has a larger magnitude. In practice, we use a voxel-grid sampling with spacing resolution $s_\text{global} = 90\,\text{mm}$ to obtain control points. We then generate random displacements of magnitude at most d_global for every control point, and interpolate them to the full point cloud using a Nadaraya–Watson estimator, parameterized by an isotropic Gaussian kernel with standard deviation σ_global.

4.a. Radius distortion. Having altered the 3D coordinates of our points using the deformations above, we add random noise to the local estimates α_i of the vessel radii – which are encoded as additional point features as detailed in Sec. A.1 and used in our RobOT layer as point weights. This additive noise is scaled by a positive parameter $s_\text{radius} = 0.1$ and drawn at random in $[-s_\text{radius}\alpha_i, s_\text{radius}\alpha_i]$.

4.b. Inconsistent sampling. By construction, the steps above let us create an arbitrary number of pairs of (real, simulated) lungs with known pairwise correspondence between all points. In order to simulate acquisition artifacts and introduce challenging inconsistencies, we sample $\mathrm{N} = \mathrm{M} = 60\text{k}$ points at random from the source and the synthetic (target) point clouds as a last generation step. We note that for training, the ground-truth flow is computed based on the source sampling and is not affected by this last degradation: it may or may not point to a sample in the synthetic target.

Augmentation of the the target and source point clouds. In our experiments, we generate our target point clouds using $d_\text{global} = 25\,\text{mm}$, $\sigma_\text{global} = 25\,\text{mm}$. Additionally, we also perform data augmentation for the source point cloud itself using the smaller values of $d_\text{global} = 8\,\text{mm}$ and $\sigma_\text{global} = 15\,\text{mm}$.



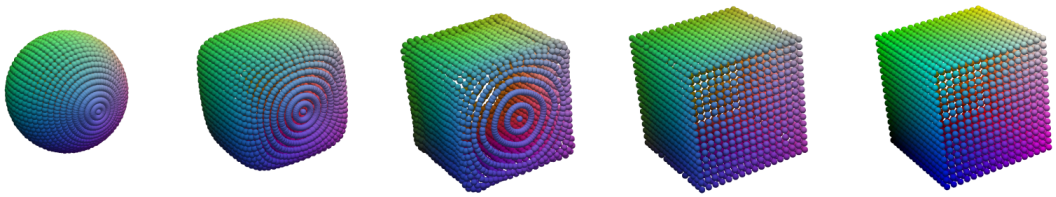
(a) Real source. (b) Local kernels. (c) Coarse control points.

(d) Synthetic target. (e) Local perturbation, source. (f) Global deformation, source.

Figure 6: **Local and global deformations that we use to generate our synthetic training dataset.** (a) Original (real) vascular tree. (b) Ellipsoids that represent anisotropic kernels of size 2 mm that we use to generate vessel-preserving local deformations. Note that in our experiments, we use larger kernels of size 4 mm that induce a stronger regularization but are harder to display cleanly. (c) Spline control points that we sample using a voxel-grid scheme with 90 mm spacing and use to generate a global deformation. (d) Synthetic vascular tree, the output of the process that we use as a target for training. (e) Source point cloud after local deformation. (f) After global deformation.

(a) Source. (b) Target. (c) FilterReg, target. (d) S-RobOT, target. (e) S-RobOT weights.

Figure 7: **Partial rigid registration** of the Stanford scans based on the matching of FPFH features – as discussed in Sec. 3.1 and Suppl. A.3.1. The feature-based CPD model FilterReg falls in a local minimum while the rigid S-RobOT of Eq. (5) results in a successful registration. In the last column, we display the attention weights w_i that are derived from unbalanced OT.



(a) Source. (b) Spline. (c) LDDMM. (d) RobOT. (e) Target.

Figure 8: **Smooth RobOT (S-RobOT) on a toy registration problem,** the deformation of a sphere (left) onto a cube (right). From left to right, we display: (a) the source shape A $= (x_1, \dots, x_N)$; (b) the output of Spline RobOT from Eq. (7); (c) the output of LDDMM RobOT, solution of the optimization problems of Eq. (8,10); (d) the output of the "raw" RobOT matching $x_i \mapsto x_i + v_i$ from Eq. (4); (e) the target shape B $= (y_1, \dots, y_M)$.

		Rotation			Translation		
	Method	**MSE** degrees^2 ↓	**RMSE** degrees ↓	**MAE** degrees ↓	**MSE** 3D units^2 ↓	**RMSE** 3D units ↓	**MAE** 3D units ↓
Matching	ICP	1134.552	33.683	25.045	0.0856	0.293	0.250
	FGR [134]	126.288	11.238	2.832	0.0009	0.030	0.008
	Go-ICP [130]	195.985	13.999	3.165	0.0011	0.033	0.012
	S-RobOT (rigid)	2.411	5.814	1.245	0.0002	0.014	0.009
End-to-end	PointNetLK [2]	280.044	16.735	7.550	0.0020	0.045	0.025
	DCP(v2) [121]	45.005	6.709	4.448	0.0007	0.027	0.020
	PRNet [122]	10.235	3.199	1.454	0.0003	0.016	0.010
	Partial-OT [28]	**0.107**	**0.328**	**0.052**	**3.384e-06**	**0.00183**	**0.0003**

Table 2: **Partial-to-Partial registration with unseen objects on ModelNet40.**

		Rotation			Translation		
	Method	**MSE** degrees^2 ↓	**RMSE** degrees ↓	**MAE** degrees ↓	**MSE** 3D units^2 ↓	**RMSE** 3D units ↓	**MAE** 3D units ↓
Matching	ICP	1217.618	34.894	25.455	0.086	0.293	0.251
	FGR [134]	98.635	9.932	1.952	0.0014	0.038	0.007
	Go-ICP [130]	157.072	12.533	2.940	0.0009	0.031	0.010
	S-RobOT(rigid)	50.997	7.142	4.012	0.0005	0.022	0.013
End-to-end	PointNetLK [2]	526.401	22.943	9.655	0.0037	0.061	0.033
	DCP(v2) [121]	95.431	9.769	6.954	0.0010	0.034	0.025
	PRNet [122]	24.857	4.986	2.329	0.0004	0.021	0.015
	Partial-OT [28]	**0.127**	**0.357**	**0.069**	**3.953e-06**	**0.002**	**0.0004**

Table 3: **Partial-to-Partial registration with unseen categories on ModelNet40.**

A.3 Additional material on S-RobOT and global feature matching

A.3.1 Partial registration

Experiment 1: Stanford scans. We now discuss a toy example, illustrated in Fig. 7, that showcases the benefits of unbalanced optimal transport theory for partial rigid registration. We normalize and resample the partial Stanford scans using a voxel-grid sampling with 0.005 spacing. We then rotate every source shape by (120°,10°,10°) degrees to create a target shape. We illustrate two methods:

- As a baseline competitor, we use the FilterReg [44] implementation of `https://github.com/neka-nat/probreg` (a feature-based CPD method [81]) with noise ratio set to 0.7 and an automatic annealing strategy based on the Expectation-Maximization (EM) algorithm.
- For our rigid S-RobOT registration, we first compute a 33-dimensional FPFH feature vector [100] for each point using a radius of 0.02 for the normal search and a radius of 0.05 for the feature search. We then rely on Eqs. (4-5) to compute the weighted RobOT matching and project it onto the space of rigid transformations. For our robust OT problem, we set the *blur* parameter to 0.1 and the *reach* parameter to 2 units in $\mathbb{R}^{33}$.

Results. Given standard FPFH features [100], our RobOT approach successfully registers the low-overlap pair while FilterReg [44] fails. The RobOT confidence weights w_i of Eq. (4) naturally act as an attention mechanism.

Experiment 2: ModelNet40. Going further, we evaluate our partial registration strategy S-RobOT on the standard dataset ModelNet-40 [127] with comparisons to state-of-the-art methods. Instead of relying on FPFH features, we use a self-supervised deep feature learning strategy to increase the robustness and accuracy of the registration. The ModelNet40 dataset contains 12,311 CAD (Computer-Aided Design) models, from 40 object categories. We follow the same experimental setup as in [28, 122]:

1. We normalize the point clouds to fit in the cube $[-1, 1]^3$.
2. To generate a registration pair, we first sample a random source shape using 1,024 points. We then apply a random rigid transformation along each axis, with rotation angle in $[0°, 45°]$ and translation in $[-0.5, +0.5]$.
3. To simulate a partial acquisition, we sample one point at random in both of the source and target shapes. In each shape, we then keep the 768 nearest neighbors of these points to define the observed regions.

Evaluation metrics. To assess the quality of our rigid registrations, we evaluate the rotation and translation errors separately. On each component, we compute the mean squared error (MSE), the root mean squared error (RMSE) and the mean absolute error (MAE).

S-RobOT setup. For feature learning, we proceed in two steps:

1. Given a source point cloud, we synthesize a target point cloud by applying a random rigid transform. For convenience, we re-use the augmentation strategy detailed above: along each axis, we sample a rotation in $[0°, 45°]$ and a translation in $[-0.5, +0.5]$.
2. We train a **self-supervised deep feature extractor** using the loss function of Sec. A.3.3. We use a PointNet++ [93] with 10,654 parameters that learns a 30-dimensional feature vector for each point.

For the S-RobOT matching, we rely on the rigid projection formula of Eq. (5); we set the *blur* parameter to $\sigma = 0.01$ and the *reach* parameter to $\tau = 10$ units in $\mathbb{R}^{30}$.

Partial-to-Partial registration with unseen objects. We follow [28, 122] and split the dataset of 12,311 point clouds into a training set with 9,843 shapes and a testing set with 2,468 shapes. We first train on **all** 9,843 shapes from **all 40 categories** in the training set and test on **all** 2,468 unseen shapes in the test set. In Table 2, we compare S-RoboT with feature matching methods (ICP, FGR [134], Go-ICP [130]) and end-to-end deep learning models (PointNetLK [2], DCP(v2) [121], PRNet [122] and Partial-OT [28]).

Partial-to-Partial registration with unseen categories. We then follow [122] and test the generalization ability of our model between object categories: we train on the first 20 categories of ModelNet40 and test on the remaining 20 categories. We report these results in Tab. 3.

Results. Overall, we observe that the unsupervised Rigid S-RobOT method performs much better than traditional approaches and is close to end-to-end methods. Since our main focus is on free-form registration, we do not push these experiments further: we use Affine and Rigid S-RobOT as pre-alignment steps and are satisfied with this level of performance.

We note that the best-performing method Partial-OT also relies on an optimal transport layer: as discussed in [105], **partial** OT is a very close cousin of the theory of **unbalanced** OT that we leverage in our RobOT layer. We understand the Partial-OT method as a supervised and end-to-end version of our S-RobOT baseline. Both [28] and this manuscript thus show that **optimal transport theory is ready to be part of the standard toolbox in our field**, with state-of-the-art performance in varied and complementary application settings.

A.3.2 Diffeomorphic registration

Background on LDDMM. Going beyond rigid and affine transformation models, the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework captures large, smooth and invertible deformations in a principled way [5]. This fluid-based model is standard in computational anatomy, especially for applications to neuroimaging where the preservation of shape topology is a key registration prior [35].

In the LDDMM model, deformations are encoded via the integration of a time-varying velocity field $v^t = \frac{\mathrm{d}}{\mathrm{d}t} x^t$ over the unit time interval $t \in [0,1]$. Given any two shapes A and B to register with each other, we seek a geodesic path $(v^t)_{t\in[0,1]}$ for a chosen Riemannian metric $\|v\|_K^2$ that is induced by a convolution kernel K_x over the space of vector fields $v : \mathbb{R}^3 \to \mathbb{R}^3$.

Following standard derivations from optimal control theory [5, 35, 54], we know that plausible deformations are fully parameterized by the **momentum** $m^0 : \mathbb{R}^3 \to \mathbb{R}^3$ at time $t = 0$, a vector field that acts as the parameter "θ" of the LDDMM deformation model. In the LDDMM framework, the optimization problem of Eq. (8) reads:

$$\theta^* = m^{0*} = \underset{m^0:\mathbb{R}^3\to\mathbb{R}^3}{\arg\min} \underbrace{\lambda_{\text{reg}} \langle m^0, K_x * m^0 \rangle_{L^2(\mathbb{R}^3,\mathbb{R}^3)}}_{\text{Regularization.}} + \underbrace{\sum_{i=1}^{\mathrm{N}} w_i \|x_i + v_i - \text{Morph}(m^0, x_i)\|^2_{\mathbb{R}^3}}_{\text{Fidelity to the data.}} , \tag{10}$$

where the deformation model Morph : $(m^0, x_i^0) \mapsto x_i^1$ is computed through the integration of the geodesic shooting equation:

$$\tfrac{d}{dt} x^t = +\tfrac{\partial H}{\partial m}\left(x^t, m^t\right), \ \tfrac{d}{dt} m^t = -\tfrac{\partial H}{\partial x}\left(x^t, m^t\right) \tag{11}$$

from time $t = 0$ to time $t = 1$ for the Hamiltonian $H(x, m) = \frac{1}{2}\langle m, K_x * m\rangle_{L^2(\mathbb{R}^3,\mathbb{R}^3)}$. We refer to Chapter 5 of [35] for a presentation and implementation of these equations on point clouds with the PyTorch and KeOps libraries.

Black-box deformation models. Different transformation models may result in different projection results. In Fig. 8, we show several smooth RobOT (S-RobOT) results for spline and LDDMM models on a toy registration task.

Experiment. The source sphere (left) and the target cube (right) are sampled with $\mathrm{N} = 1,922$ and $\mathrm{M} = 1,538$ points respectively. We normalize their coordinates to be contained in $[-1,1]^3$ and normalize the point cloud weights to sum up to one ($\alpha_i = 1/1,922$ and $\beta_j = 1/1,538$ so that $\sum_{i=1}^{1,922} \alpha_i = \sum_{j=1}^{1,538} \beta_j = 1$). For the initial RobOT matching, we set the *blur* parameter to 0.005 units in $\mathbb{R}^3$ and the *reach* parameter to $+\infty$ (balanced OT). For the following spline and LDDMM regularizations, we use a multi-Gaussian-kernel with standard deviations $\{0.05, 0.2, 0.3\}$ and weights $\{0.2, 0.3, 0.5\}$, i.e. a kernel function:

$$\begin{aligned} k(x,y) \ &= \ 0.2 \cdot \exp\left[-\|x-y\|^2_{\mathbb{R}^3}/(2\cdot 0.05^2)\right] \ + \ 0.3 \cdot \exp\left[-\|x-y\|^2_{\mathbb{R}^3}/(2\cdot 0.2^2)\right] \\ &+ \ 0.5 \cdot \exp\left[-\|x-y\|^2_{\mathbb{R}^3}/(2\cdot 0.3^2)\right] . \end{aligned} \tag{12}$$

Results. We make two important observations:

- As evidenced by the disappearance of the **polar sampling pattern** in the fourth column, the raw RobOT matching $x_i \mapsto x_i + v_i$ provides a "perfect fit" to the target but does not preserve the topology of the source point cloud. The spline and the LDDMM approximations alleviate this problem: they smooth the weighted RobOT matching to combine accuracy with topology preservation and, in the case of the LDDMM model, guarantees of invertibility.
- The spline and the LDDMM model both result in smooth deformations. But crucially, the LDDMM model is more suited to large deformations and provides a **better fit to the corners of the target cube**. We note that workarounds exist for the over-smoothing of the second (spline) column: at the cost of an increased sensitivity to noise, singular kernels [109] or ridge regression methods [10] allow spline models to get a closer fit to the target. For medical applications, the key benefit of LDDMM is that it provides strong guarantees on the invertibility of the deformation: we refer to Chapter 5 of [35] for a detailed discussion. In the remainder of this work, we benchmark both spline and LDDMM deformation models whenever relevant. We observe that diffeomorphic LDDMM registrations perform better on e.g. lung data, but stress that such comparisons are task-dependent.

A.3.3 Deep feature learning

Feature learning on 3D point clouds. As discussed above, (x, y, z) coordinates or standard FPFH features can be good enough to handle simple shapes and deformations. On challenging settings however, learning task-specific features is often key to high performance.

In Sec. 3.2 and Suppl. A.4, we present our best-performing solution to this problem: the D-RobOT architecture. In this section, we discuss an **alternative approach** that decouples feature learning and matching. In practice, we did not obtain competitive results with this method. Nevertheless, using a separate feature extractor may increase interpretability and ease deployment issues in medical scenarios: we believe that this line of work is worth pursuing and provide full details on our experiments.

Let us process a synthetic pair of point clouds with N points in correspondence with each other – $\mathrm{N} = 60,000$ in our experiments for lung registration. Instead of sampling positive and negative samples, which is common for contrastive loss functions, we choose to rely on $\mathrm{N} \times \mathrm{N}$ correspondence matrices that take **all possible point pairs into account**. We use the KeOps library [37] to manipulate these objects efficiently, with extremely fast run times and without memory overflows.

Specifically, we train our feature extraction network as follows:

1. **Input data.** We assume that we are given two point clouds $(x_1, \dots, x_{\mathrm{N}})$ and $(y_1, \dots, y_{\mathrm{N}})$ that are in pairwise correspondence with each other. In our experiments, these are typically the output of a synthetic "data augmentation" procedure: the Flying3D objects for the Kitti benchmark and our synthetic lung pairs for the Dirlab benchmark.
2. **Feature extraction.** We apply the feature extractor (a trainable point neural network) on both point clouds, independently from each other. We retrieve point features p_i and q_i in $\mathbb{R}^{\mathrm{D}}$ that are respectively associated to the points x_i and y_i.
3. **Feature normalization.** As discussed in Sec. 2.1, we normalize the feature vectors so that $\|p_i\|_{\mathbb{R}^{\mathrm{D}}} = \|q_i\|_{\mathbb{R}^{\mathrm{D}}} = 1$. This prevents the feature extractor from converging to degenerate solutions and ensures that our hyper-parameters for the RobOT problem of Eq. (2) can be interpreted as sensible scales in $\mathbb{R}^{\mathrm{D}}$.
4. **Source self-similarities.** We compute the $\mathrm{N} \times \mathrm{N}$ correspondence matrix for the point positions in the source point cloud:

$$c_{(x_i,x_j)} = \mathrm{softmax}_{j=1}^{\mathrm{N}}(-\|x_i - x_j\|^2_{\mathbb{R}^3}/2\kappa^2) = \frac{\exp(-\|x_i - x_j\|^2_{\mathbb{R}^3}/2\kappa^2)}{\sum_{j=1}^{\mathrm{N}} \exp(-\|x_i - x_j\|^2_{\mathbb{R}^3}/2\kappa^2)} \,, \quad (13)$$

where the Softmax denotes a mirrored exponential followed by a normalization over the rows of the correspondence matrix while $\kappa > 0$ is a scaling factor. Each row of the matrix $c_{(x_i,x_j)}$ then refers to a probability distribution, a position heatmap with a peak at point x_i whose radius is proportional to κ.

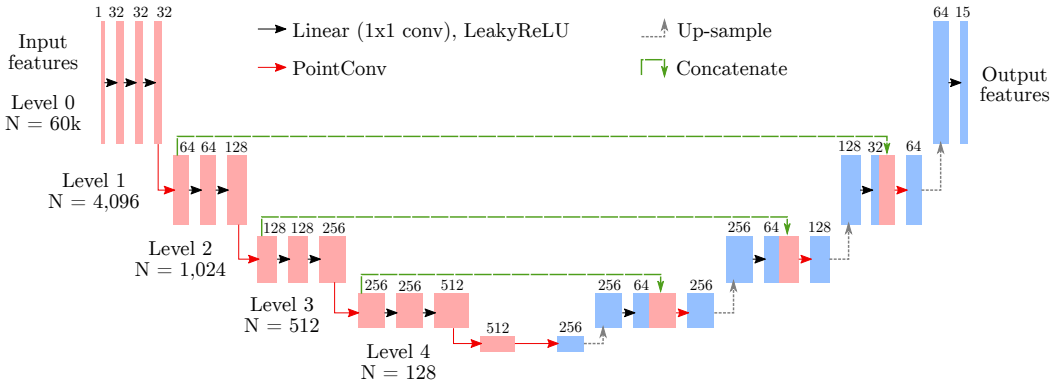


Figure 9: **Deep feature extractor** for our S-RobOT experiments. We adapt this feature pyramid network from PointPWC-Net [126]. We implement a four-scale U-net [95] architecture using: point-wise multi-layer perceptrons on the feature embeddings (with LeakyReLU non-linearities); PointConv [125] for downsampling and gathering information from the neighbors; K-Nearest Neighbor interpolation for upsampling.
Please note that this architecture relies on the point coordinates (x, y, z) to define the PointConv convolution and the upsampling layer. In our experiments for lung registration, we use the local vessel radius as our only input feature: this corresponds to using a single input channel on the left-most layer of the figure above.

5. **Feature correspondences.** Similarly, we compute a correspondence matrix for the source and target features:

$$c_{(p_i,q_j)} = \text{softmax}_{j=1}^{\text{N}}(-\|p_i - q_j\|_{\mathbb{R}^{\text{D}}}^2) \,. \tag{14}$$

   Each row of $c_{(p_i,q_j)}$ is a feature heatmap that indicates how well p_i corresponds to q_j.
6. **Training loss.** Our total loss is the sum over the cross entropies for each row:

$$\text{CE}(c_{(x_i,x_j)}, c_{(p_i,q_j)}) = -\sum_{i=1}^{\text{N}}\sum_{j=1}^{\text{N}} c_{(x_i,x_j)} \log\ c_{(p_i,q_j)} \,. \tag{15}$$

   We optimize it by stochastic gradient descent over the parameters of the feature extraction network (step 2).

Hyper-parameters. For feature learning, we use a U-net structured PointConv [125] architecture with 7M parameters that is illustrated in Fig. 9. At the feature learning stage, we set κ (described above) to $\sqrt{2}$ mm. For each point we learn a 15 dimensional feature vector of unit length. For the RobOT feature matching, we set the *blur* parameter to $\sigma = 0.01$ units in $\mathbb{R}^{15}$ and the *reach* parameter to $\rho = +\infty$ (balanced OT). As discussed in Suppl. A.4.2, we benchmark two types of S-RobOT regularization:

1. The spline smoothing of Eq. (7), using a Gaussian (RBF) kernel with a standard deviation of 5 mm.
2. An LDDMM deformation model that we optimize as in Eq. (8) using an SGD solver. We use a multi-Gaussian-kernel with standard deviations $\{5, 8, 10\}$ mm and weights $\{0.2, 0.3, 0.5\}$.

A.4 Additional material on deep deformation prediction

We now provide full details on our D-RobOT architecture and additional experiments on scene flow estimation and lung registration.

A.4.1 Deep registration module

Architecture of the prediction network. In Sec. 3.2, we rely on a modified PointPWC-Net to act as a predictor $\text{Pred} : (x_i, y_j) \mapsto \theta$. This multiscale point neural network takes as input the source and target point clouds. It returns a high-dimensional parameter θ for the deformation model $\text{Morph} : (\theta, x_i) \mapsto \hat{y}_i$. In this work, the predicted θ is always a 3D vector field that is supported by

the points x_i (for the raw displacements model) or by a collection of control points c_i that have been generated by farthest point sampling (for the spline and LDDMM models).

We describe our modifications to the original PointPWC-Net architecture in Fig. 10. In order to define a network that can predict spline and LDDMM parameters:

1. We replace the flow prediction layer of PointPWC-Net by a suitable prediction layer for registration parameters.
2. We use an asymmetric hierarchical architecture that outputs registration parameters for the control points c_i. Since the number of control points is often much smaller than the number of points in the input point cloud, this reduces the memory footprint of our network architecture.

Losses. We use synthetic data pairs for training: for scene flow estimation on Kitti, we rely on the synthetic Flying3D dataset; for lung registration, we rely on the two-scales simulator of Suppl. A.2. In both cases, we thus have access to ground truth deformations and can rely on a supervised learning strategy. Let us consider a PointPWC-Net with L scales, and denote by $(x_1^l, \dots, x_{\mathrm{N}_l}^l)$ the subsampled input for the l-th scale. For training, we compute a multi-scale similarity loss as:

$$\mathrm{Loss}(\hat{y}_i) \;=\; \sum_{l=0}^{\mathrm{L}-1} W^l \sum_i w_i \cdot \| \, \hat{y}_i^l \;-\; y_i^l \, \|_2^2 \,, \tag{16}$$

where $l \in \{0, \dots, \mathrm{L}\}$ is a scale ($l = 0$ corresponds to the raw point clouds), W^l is a scalar hyper-parameter that we use as a total weight for the l-th scale, $\hat{y}_i^l$ is the output of the network that corresponds to the flowed x_i^l at the l-th scale and y_i^l denotes the ground truth target that corresponds to the same point x_i^l with weight w_i. Note that at the finest scale, we compute $\hat{y}^0$ using a task-specific deformation model, Morph : $(\theta, x_i) \to \hat{y}_i$.

A.4.2 Experiments on lung vascular trees

We now provide more details on our experiments for lung registration, which are illustrated in Fig. 11 with a visualization of landmark errors in Fig. 13. Notably, we discuss both supervised and unsupervised training strategies as well as the influence of the deformation model Morph : $(\theta, x_i) \mapsto \hat{y}_i$ on the D-robot architecture.

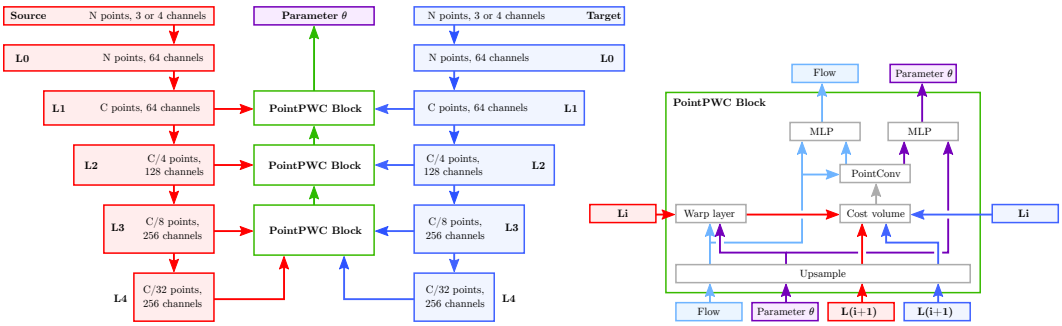
Synthetic data vs unsupervised learning. To overcome the lack of dense 3D annotations on real shape data, we advocate the use of simulated deformations and synthetic training datasets. As detailed in Sec. 3.2, we observe that D-RobOT networks are easy to train to a high level of accuracy: RobOT-based post-processing and fine-tuning help our models to bridge the domain gap between the synthetic and real distributions of shapes.

We would like to stress that our focus on synthetic training datasets to the detriment of e.g. unsupervised approaches results from **careful and extensive experiments**. In the lead-up to the publication of this work, we tried several competing approaches to train our registration networks: supervised learning with dense correspondences on synthetic data; "unsupervised" learning on unannotated point clouds using geometric loss functions between point sets; a mix of both approaches. In practice, supervised learning on synthetic data clearly emerged as the most practical option for challenging registration tasks. Let us briefly explain why.

Unsupervised loss functions. When dense pointwise correspondences are not available, a popular strategy to train registration methods is to rely on permutation-invariant loss functions between point clouds. We refer to Chapters 3 and 4 of [35] for an introduction to the topic.

Since our PVT1010 dataset contains 1,000 pairs of lung vessel trees that are in correspondence with each other (inspiration/expiration) but for which no expert-annotated landmarks are available, we are in a perfect situation to try out these tools. We thus attempted to train our networks using local Laplacian matching [126], Maximum Mean Discrepancies, Gaussian Mixture Models (GMM) as well as Wasserstein distances [35] on (x, y, z) coordinates. Unfortunately, we **never succeeded in converging to a competitive level of accuracy**. We believe that this is due to the **complex geometric structure of the lung vascular trees**, which are significantly more intricate than the clean point clouds and surface meshes on which these methods are usually tested [36, 40].

Combining synthetic (supervised) and real (unsupervised) data. Going further, we also tried to use a **mixed strategy**: in our training dataset, we combined synthetic deformations of real source shapes (that can be handled using a mean square error) with genuine target point clouds (that can be



(a) Multiscale prediction architecture.

(b) Architecture of a PointPWC block.

Figure 10: **Architecture of our deep prediction network** Pred : $(x_i, y_j) \mapsto \theta$, which is adapted from PointPWC-Net [126]. The original PointPWC-Net architecture is a multi-scale network that predicts 3D scene flow (a raw displacement field) in a coarse-to-fine fashion. We modify this state-of-the-art architecture to make it suitable for **registration with a task-specific deformation model** Morph : $(\theta, x_i) \mapsto \hat{y}_i$.
a) First of all, we compute **feature pyramids** at scales L0 to L4 using farthest point sampling and PointConv layers [125] on the source and target shapes (left and right columns) – see Fig. 5 in [126]. In our experiments, we always use $\mathrm{C} = 4,096$ or $\mathrm{C} = 8,192$ **control points**. The key design decision behind the PointPWC-Net architecture is to run through this pyramid from the coarsest scale (L4) to the finest one (L0, the original point clouds) in order to predict the final 3D scene flow: we represent this coarse-to-fine prediction as a stack of **PointPWC blocks** (central column).
b) Those blocks share the same architecture (but not the same neural weights): an **upsampling** layer that interpolates our 3D vector fields from the coarser to the finer scale using an inverse distance spline kernel; another **upsampling** layer that interpolates the pyramid features at the coarsest scale; a **warp layer** that deforms the source shape according to the up-sampled 3D flow, allowing the network to focus on **residual** deformations at each scale; a **cost "volume"** layer, detailed in Fig. 2 of [126], which relies on K-NN neighborhoods and PointConv layers to merge the source and target features into a vector of "patchwise" discrepancies for every point in the (subsampled and warped) source shape; a **prediction** PointConv and Multi-Layer Perceptrons (MLP), detailed in Fig. 6 of [126], that turns this vector of discrepancies into a predicted correction for the up-sampled 3D flow.
Please note that this architecture relies on the point coordinates (x, y, z) in the PointConv and upsampling layers. For our scene flow experiments, we use the (x, y, z) coordinates as input to the network: this corresponds to "Source" and "Target" arrays that have 3 input channels. For our lung registration experiments, we also add the local vessel radius as a fourth feature and thus use 4 input channels. We stress that in our RobOT-based pre-alignment and post-processing (steps 1 and 3 of the D-RobOT architecture), we rely on the (x, y, z) coordinates as point features p_i and q_j in Eq. (2). When available, the vessel radii are only used as point weights α_i and β_j in the RobOT problem.
In all our experiments, the parameter θ is a 3D vector field that has the same memory footprint as a "raw" 3D scene flow supported by our control points: this corresponds to using 3 output channels on the C control points. Our main modification to the original PointPWC-Net architecture is to **handle this registration parameter** (purple) **in parallel with the usual scene flow** (sky blue): the final prediction module of every PointPWC block runs **two estimations in parallel**, which share the same PointConv layer and are equivalent to Fig. 6 in [126].

handled using e.g. the Wasserstein distance). We expected that this combination would narrow the **domain gap** between our synthetic deformations and the real breathing movement. In practice, this strategy produced **indecisive results**:

- On the one hand, assuming that we only have access to a **simple deformation model** (e.g. a single-scale random field), this mixed training strategy improves the accuracy of our registrations. This is especially true when the synthetic deformations are not diverse enough.
- On the other hand, assuming that we have access to the more **realistic and expressive** deformation model of Suppl. A.2, the introduction of real but not annotated pairs in the training loop proves **slightly detrimental** to performance. We observe a small increase of about 0.2mm in landmark Root Mean Squared Error (RMSE).

For the sake of **simplicity**, we thus trained our final D-RobOT network entirely on synthetic deformations. We found that using RobOT as a **final post-processing** layer in the D-RobOT pipeline is enough to address prediction errors effectively and compensate for a small domain gap between synthetic and real deformations.

Raw displacements vs spline and LDDMM regularizations. In Fig. 12, we further compare the D-RobOT results for three different deformation models. Our main observations are that:

- A deep prediction module that returns raw displacements Δx_i with a naive deformation model $\text{Morph} : (\theta = \Delta x_i, x_i) \mapsto \hat{y}_i = x_i + \Delta x_i$ may produce non-smooth registration results. This holds even when the network is trained entirely on smooth, synthetic deformations. On the other hand, the regularizing spline and LDDMM models always result in smooth deformations. This vindicates the use of **explicit regularizing models** in safety-critical applications: we cannot trust our networks to "learn smoothness properties" from the data.
- According to Fig. 12 and Tab. 4, the LDDMM model captures large deformations better than the spline model. However, we obtain similar performance with both spline and LDDMM models after the (affordable) RobOT post-processing step: **fine-tuning with RobOT alleviates the need for complex and expensive deformation models**.

Experiments. We now provide detailed hyperparameters for the D-RobOT architecture in our lung registration experiments. As detailed in Sec. 3.2, the full model applies successively: an affine S-RobOT pre-alignment; a deep non-parametric registration; and a spline S-RobOT post-processing.

1. Affine registration. For affine pre-alignment, we use the S-RobOT model of Eq. (6) with raw (x, y, z) coordinates as input features in $\mathbb{R}^3$. The vessel radii are taken into account as point weights α_i and β_j. In the RobOT problem of Eq. (2), we set the *blur* parameter to $\sigma = 1$ mm and the *reach* parameter to $\tau = +\infty$ (balanced OT with strong constraints on the marginals).

2. Deep non-parametric registration. We evaluate three deformation models $\text{Morph} : (\theta, x_i) \mapsto \hat{y}_i$:

1. **Raw displacements** correspond to the case where the parameter $\theta = \Delta x_i$ is a 3D vector field that is supported by the point x_i and the deformation model is the addition:
$$\text{Morph} \ : \ (\Delta x_i, x_i) \in \mathbb{R}^{\mathrm{N}\times 3} \times \mathbb{R}^{\mathrm{N}\times 3} \ \mapsto \ x_i + \Delta x_i \in \mathbb{R}^{\mathrm{N}\times 3} \, . \tag{17}$$
As a regularization penalty, we use the squared Euclidean norm:
$$\text{Reg}(\Delta x_i) = \tfrac{1}{\mathrm{N}} \textstyle\sum_{i=1}^{\mathrm{N}} \|\Delta x_i\|^2_{\mathbb{R}^3} \, . \tag{18}$$
2. The **spline** model corresponds to the case where the parameter $\theta = \Delta c_l$ is a 3D vector field that is supported by a set of control points $(c_1, \dots, c_{\mathrm{C}})$ in $\mathbb{R}^3$. As a deformation model, we use a simple kernel smoothing:
$$\text{Morph} \ : \ (\Delta c_l, x_i) \in \mathbb{R}^{\mathrm{C}\times 3} \times \mathbb{R}^{\mathrm{N}\times 3} \ \mapsto \ x_i + \textstyle\sum_{l=1}^{\mathrm{C}} k(x_i, c_l) \Delta c_l \in \mathbb{R}^{\mathrm{N}\times 3} \, . \tag{19}$$
As a regularization penalty, we use the squared Euclidean norm:
$$\text{Reg}(\Delta c_l) = \tfrac{1}{\mathrm{C}} \textstyle\sum_{l=1}^{\mathrm{C}} \|\Delta c_l\|^2_{\mathbb{R}^3} \, . \tag{20}$$
For our lung experiments, we always use a multi-Gaussian kernel with standard deviations $\{3, 6, 9\}$ mm and weights $\{0.2, 0.3, 0.5\}$. With the coordinates of points x and y in millimeters, this corresponds to the positive definite kernel function:
$$\begin{aligned} k(x,y) \ = \ & 0.2 \cdot \exp\big[- \|x-y\|^2_{\mathbb{R}^3} / (2 \cdot 3^2) \big] \ + \ 0.3 \cdot \exp\big[- \|x-y\|^2_{\mathbb{R}^3} / (2 \cdot 6^2) \big] \\ & + \ 0.5 \cdot \exp\big[- \|x-y\|^2_{\mathbb{R}^3} / (2 \cdot 9^2) \big] \, . \end{aligned} \tag{21}$$

3. The **diffeomorphic LDDMM** model is equivalent to a spline deformation with continuous time. Notably, we use the same kernel $k(x, y)$ as in the spline model above. As detailed in Suppl. A.3.2 and Chapter 5 of [35], the parameter $\theta = m_l^0$ is a 3D vector field that is supported by a set of control points $(c_1, \dots, c_{\mathrm{C}})$ in $\mathbb{R}^3$. As a deformation model, we use an iterative kernel smoothing and deformation layer that corresponds to a numerical integration scheme for the Ordinary Differential Equation (11). More specifically, we use the differentiable Dopri-5 layer that is provided by the TorchDiffEq library [22, 23], which implements the Dormand-Prince or Runge-Kutta 4(5) integrator [32] with 20 time steps from time $t = 0$ to time $t = 1$. For a simpler implementation, we refer to the Euler integration scheme that is detailed in [35], Algorithm 5.6. As a regularization term, we use the squared kernel norm:

$$\mathrm{Reg}(\Delta c_l) = \tfrac{1}{\mathrm{C}} \sum_{l=1}^{\mathrm{C}} \sum_{s=1}^{\mathrm{C}} k(c_l, c_s) \langle m_l^0, m_s^0 \rangle_{\mathbb{R}^3} \ . \tag{22}$$

For the **deep prediction network** $\mathrm{Pred} : (x_i, y_j) \mapsto \theta$, we use the 4-scale hierarchical architecture that is described in Suppl. A.4.1 and set the scale weights W^l to $\{1.0, 0.8, 0.4, 0.2\}$. We compute the 4-scale decompositions of the point clouds using Farthest Point Sampling. For all models, we promote a close fit to the target shape and weight the regularization term $\mathrm{Reg}(\theta)$ by 1/100 in the training loss:

$$\mathrm{Loss}(x_i, y_i, \theta) \ = \ \tfrac{1}{100}\mathrm{Reg}(\theta) \ + \ \underbrace{\sum_{l=0}^{\mathrm{L}-1} W^l \sum_i w_i \cdot \| \, \hat{y}_i^l(\theta, x_i) \ - \ y_i^l \, \|_{\mathbb{R}^3}^2}_{\text{Multiscale loss of Eq. (16)}} \ . \tag{23}$$

We recall that in the expression above:

1. The weights w_i are attached to the points x_i and are proportional to the vessel radius.
2. The weights W^0, W^1, W^2 and W^3 put a strong emphasis on the finest scales.
3. Each point y_i^l corresponds to the ground truth target for x_i at scale l.
4. Each point $\hat{y}_i^l(\theta, x_i)$ corresponds to the output of our prediction and deformation networks at scale l for x_i. As detailed in Fig. 10, the finest-scale output $\hat{y}_i^0(\theta, x_i) = \mathrm{Morph}(\theta, x_i)$ corresponds to our model-based deformation; the larger-scale predictions $\hat{y}_i^1(\theta, x_i)$, $\hat{y}_i^2(\theta, x_i)$ and $\hat{y}_i^3(\theta, x_i)$ correspond to raw displacements and are only used in Eq. (23) to make the training easier [126].

3. Postprocessing. To fine-tune our registration, we use the spline S-RobOT deformation of Eq. (7) with (x, y, z) coordinates as input features in $\mathbb{R}^3$. We set the *blur* parameter to $\sigma = 0.1$ mm and the *reach* parameter to $\tau = 10$ mm. For smoothing, we use the vessel-preserving anisotropic kernel $k(x, y)$ of Sec. A.2 with a kernel scale $s_{\text{local}} = 8$ mm. In order to get a closer fit to the target, we apply this post-processing step twice for all lung registration results.

Post-processing	D-RobOT (raw) mm ↓	D-RobOT (spline) mm ↓	D-RobOT (LDDMM) mm ↓
Without post-processing	3.46 (3.13)	4.45 (4.10)	3.33 (3.09)
NN projection	3.33 (2.96)	3.32 (3.05)	3.31 (2.96)
RobOT	3.35 (3.04)	3.18 (3.01)	3.19 (3.01)
NN projection + Smoothing	**3.32 (2.95)**	3.08 (2.82)	2.95 (2.64)
RobOT + Smoothing	3.33 (2.99)	**2.94 (2.71)**	**2.83 (2.40)**

Table 4: **Ablation study on the post-processing module.** We assess the influence of the fine-tuning step in the D-RobOT model. As an addendum to Tab. 1, we display the Root Mean Squared Error in millimeters (median of the 10 subjects in parentheses) for expert-annotated landmarks on the 10 Dirlab lung pairs, that we use as a test set for PVT1010. The first row corresponds to the output of the deep registration module, without any fine-tuning. The second row corresponds to a nearest neighbor projection on the target point cloud. The third row corresponds to the "raw" RobOT matching of Eq. 4, extrapolated to the landmarks using a Nadaraya-Watson spline with an isotropic Gaussian kernel of deviation $\sigma = 0.5$ mm. In the fourth and fifth rows, we use a vessel-preserving anisotropic kernel to smooth a nearest neighbor projection (fourth row) and our RobOT matching (fifth row).

Method	Pre-alignment	Deep prediction	Post-processing	Number of points	Number of control points	RMSE mm ↓
Input data		None		60k	—-	23.32
Affine	✓	None		60k	—-	10.31
FlowNet3D		FLOW		30k	4,096	8.20
	✓	FLOW		30k	4,096	7.08
	✓	FLOW	✓	30k	4,096	6.51
D-RobOT	✓	PWC*		30k	4,096	4.10
		PWC*		60k	4,096	4.64
	✓	PWC*		60k	4,096	3.82
	✓	PWC*		60k	8,192	3.33
	✓	PWC*	✓	60k	8,192	**2.83**

Table 5: **Ablation study for the D-RobOT model (with LDDMM deformations) on the lung registration task.** We benchmark the Root Mean Squared Error on the 10×300 pairs of DirLab landmarks for a wide range of configurations. We investigate the influence of affine S-RobOT pre-alignment (column 2); spline S-RobOT fine-tuning (column 4); the sampling rate for the input data (column 5); and the number of control points for the LDDMM deformation model (column 6). As a backbone architecture for the deep prediction network, we use a FlowNet3D (rows 3-5) and the modified PointPWC-Net of Fig. 10 (rows 6-10).

	Method	Prealign	Post	EPE3D cm ↓	Acc3DS % ↑	Acc3DR % ↑	Outliers3D % ↓	EPE2D px ↓	Acc2D % ↑
8192 points	PWC [126]			6.78	78.58	90.30	23.61	2.6484	82.36
	PWC	✓		4.62	**83.24**	94.91	**19.61**	2.1411	86.82
	PWC	✓	✓	4.76	82.16	94.53	20.10	2.0716	**87.59**
	D-RobOT	✓		4.63	79.13	**95.30**	21.43	2.3930	82.93
	D-RobOT	✓	✓	**4.55**	80.27	94.85	20.65	2.1404	85.24
30k points	PWC [126]			7.90	61.33	89.75	29.36	4.0558	65.35
	PWC	✓		5.94	72.62	92.26	25.79	3.1817	72.82
	PWC	✓	✓	3.99	86.44	95.02	18.59	1.9592	86.09
	D-RobOT	✓		2.94	92.86	98.59	16.05	1.5733	91.86
	D-RobOT	✓	✓	**2.15**	**95.74**	**98.96**	**12.93**	**1.1118**	**95.66**

Table 6: **Ablation study on the modules of D-RobOT (with spline deformations)** and the influence of the point sampling rate, performed on the **57 largest scenes from the Kitti dataset**. As detailed in Suppl. A.4.3, these results correspond to average values on the 57 (out of 142) pairs of 3D scenes that are sampled with more than 30k points per frame in the original Kitti dataset. This allows us to focus our evaluation on the most challenging 3D scenes, with under-sampling artifacts: as expected, performance is lower than in the "full" Kitti benchmark of Fig. 4.

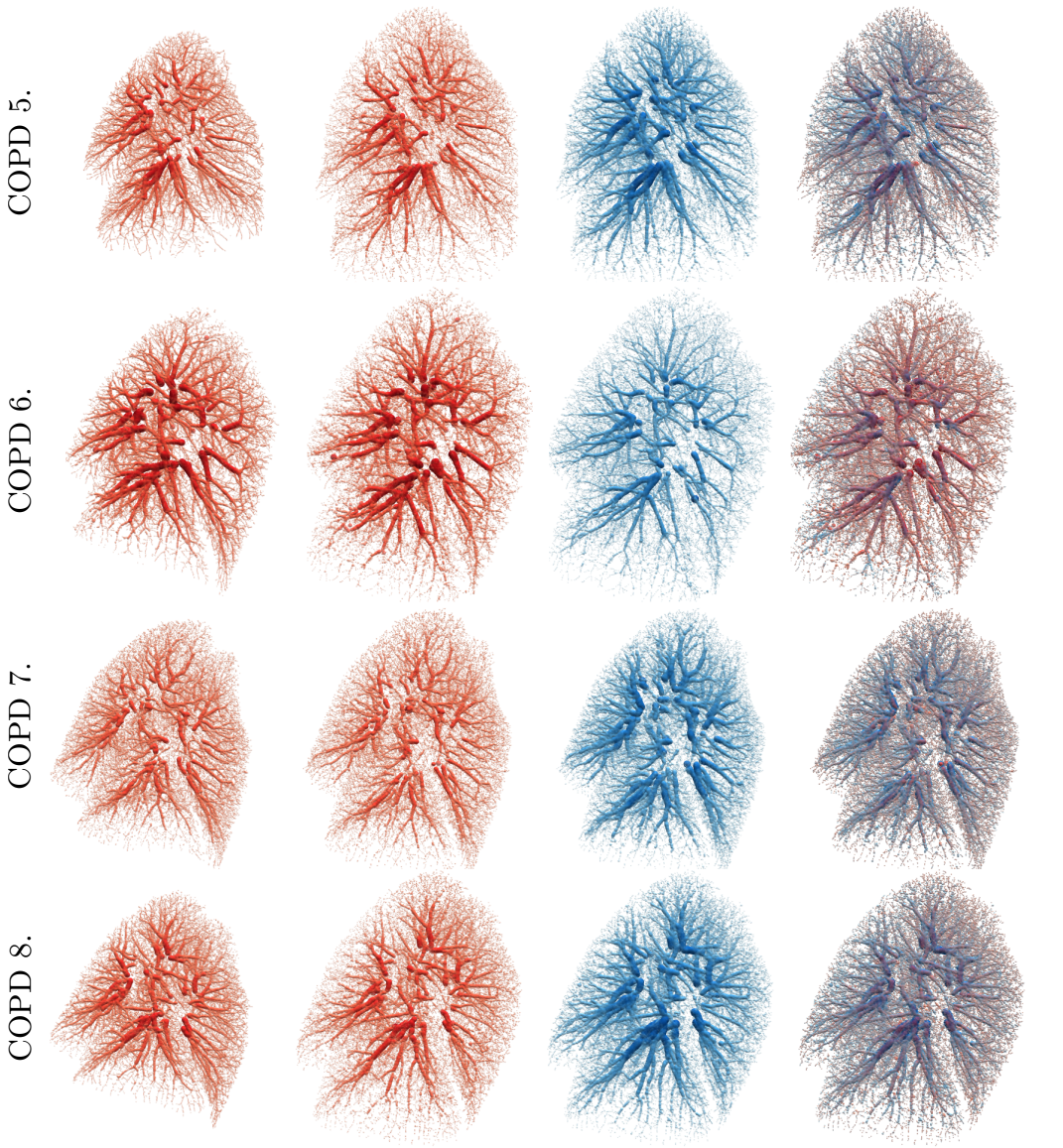
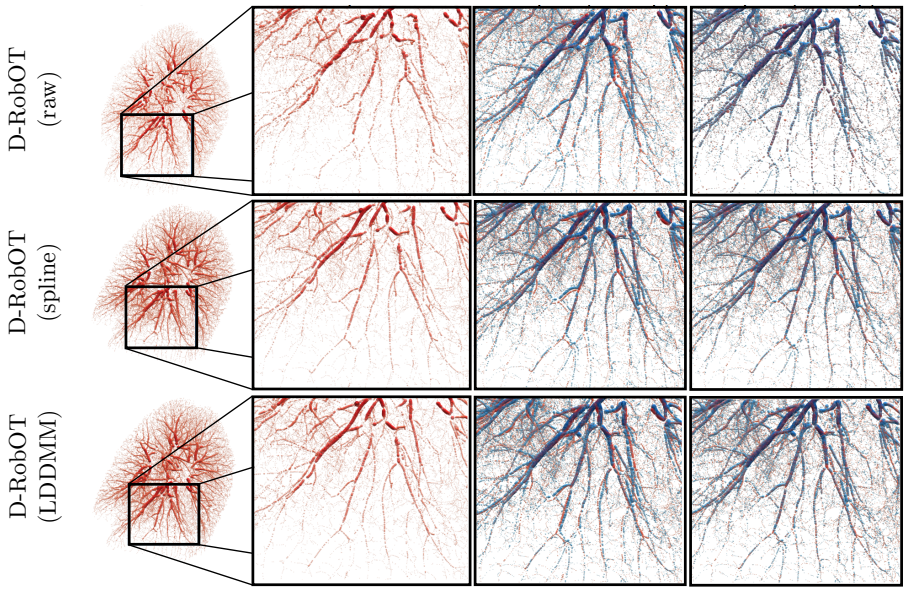
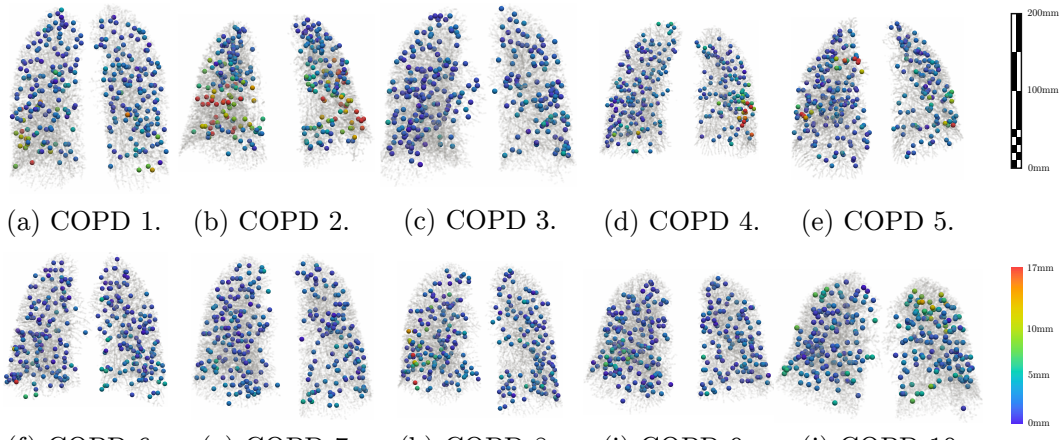


Figure 11: **Additional results for lung vascular trees.** Columns refer to: (a) the source shape (expiration); (b) the registration result from D-RobOT using the LDDMM deformation model; (c) the target shape (inspiration); (d) an overlap between the D-RobOT registration and the target. Each row corresponds to a patient, with names that refer to case IDs in the original DirLab dataset.



(a) Output of the deep registration (step 2). (b) Output of step 2, target. (c) Output of step 3, target.

Figure 12: **D-RobOT results for different deformation models Morph** : $(\theta, x_i) \mapsto \hat{y}_i$**.** (a) The first two columns correspond to the output of our deep registration module (without post-processing); (b) the third column also includes the target shape, displayed as a blue point cloud; (c) the fourth column showcases our final result, with RobOT-based post-processing. As detailed in Suppl. A.4.2, we observe that: (i) the spline and LDDMM models produce smooth registration results; (ii) before the RobOT post-processing step, LDDMM provides a better fit than the spline model; (iii) after post-processing, all three deformation models produce "sharp" results, with smoothness guarantees for the spline and LDDMM models.



(a) COPD 1. (b) COPD 2. (c) COPD 3. (d) COPD 4. (e) COPD 5.

(f) COPD 6. (g) COPD 7. (h) COPD 8. (i) COPD 9. (j) COPD 10.

Figure 13: **Distributions of the DirLab-COPDGene errors for the 300 expert-annotated corresponding landmarks.** Each image refers to a patient, with a title that refers to the case ID in the original DirLab dataset. We use color to display the registration error with our best performing model (D-RobOT with LDDMM deformations) on the source landmarks. The COPD 2 patient is a clear outlier in our experiment as in all previous studies on the original volumetric data (`https://www.dir-lab.com/Results.html`) – this may be due to annotation errors. On the other patients, most of the large errors occur in boundary regions where acquisition artifacts induce inconsistencies between the source and target point clouds: small vessels are harder to catch in the original 3D volume.

A.4.3 Experiments on Kitti

We now provide details for our experiments on scene flow estimation, which are illustrated in Fig. 14. Notably, we discuss the influence of the sampling rate for the source and target point clouds and provide an extensive ablation study for the modules of the D-RobOT model.

Influence of the sampling rate. We follow the pre-processing strategy of PointPWC-Net [126]:

1. We train on points whose depth with respect to the acquisition device is smaller than 35 m.
2. We sample points from both of the source and target frames at random, in a non-corresponding manner.

In the main manuscript, we report results when the source and target point clouds are sampled with either 8,192 or 30k points at a time. We note that in a similar situation, when sampling 32,768 points from each frame, HPLFlowNet [49] reported an average EPE3D value of 10.87 cm over all 142 pairs of the Kitti dataset. In comparison, our D-RobOT model reaches a much higher accuracy (EPE3D $=$ 2.23 cm) with only 30,000 points per frame.

Working with the 57 largest scenes from Kitti. We note that the number of points per 3D frame varies widely in the Kitti dataset. To investigate the potential negative effects from under-sampling, we evaluate our models separately on pairs where both frames contain more than 30k points: 57 out of the 142 Kitti scene pairs meet this requirement. When dealing with those 57 "most populated" frames, sub-sampling the scene to 30k points induces a net loss of geometric information.

We display our results for this subset of Kitti in Tab. 6: unsurprisingly, performance is lower than in Fig. 4, which reports results on the full Kitti dataset and thus includes the 85 pairs of frames for which 30k points are enough to work at full resolution. Nevertheless, our conclusions remain consistent: the D-RobOT method outperforms PointPWC-Net [126] for both sampling rates.

Ablation study. Going further, we analyze the contributions of the pre-alignment and post-processing modules in the D-RobOT model. Our results are summarized in Tab. 6 and can be described as follows:

- The **pre-alignment module** improves results for **both sampling rates**. Since rigid transformations have few degrees of freedom, the estimation of Eq. (5) does not rely on fine details.
- In sharp contrast, the **post-processing module** only improves results for **high-resolution points**. In practice, our RobOT-based post-processing behaves as a fast local fitting to the target point cloud. In the upper half of the table, the target point cloud and the set of "ground truth" final positions for the source points have small overlap due to the small number of sampling points for these complex scenes: a local "pixel-perfect" fine-tuning is detrimental to performance. In the lower half of the table, we increase the sampling rate to 30k points per frame: the target point cloud and the set of ground-truth final positions for the source points have a much higher overlap. This allows the RobOT post-processing to increase the accuracy of the full model.

Experiments. We now provide detailed hyperparameters for the D-RobOT architecture in our Kitti experiments. As detailed in Sec. 3.2, the full model applies successively: a rigid S-RobOT pre-alignment; a deep non-parametric registration; and a spline S-RobOT post-processing.

1. Rigid registration. For rigid pre-alignment, we use the S-RobOT model of Eq. (5) with raw (x, y, z) coordinates as input features in $\mathbb{R}^3$. The vessel radii are taken into account as point weights α_i and β_j. In the RobOT problem of Eq. (2), we set the *blur* parameter to $\sigma = 1$ m and the *reach* parameter to $\tau = +\infty$ (balanced OT with strong constraints on the marginals).

2. Deep non-parametric registration. We use the prediction architecture and deformation models of Suppl. A.4.2. Our spline deformation model is based on a multi-Gaussian-kernel with standard deviations $\{20, 40, 60\}$ cm as and weights $\{0.2, 0.3, 0.5\}$. With the coordinates of points the x and y in centimeters, this corresponds to the positive definite kernel function:

$$\begin{aligned} k(x,y) \;=\; & 0.2 \cdot \exp\big[-\|x-y\|^2_{\mathbb{R}^3}/(2\cdot 20^2)\big] \;+\; 0.3 \cdot \exp\big[-\|x-y\|^2_{\mathbb{R}^3}/(2\cdot 40^2)\big] \\ & +\; 0.5 \cdot \exp\big[-\|x-y\|^2_{\mathbb{R}^3}/(2\cdot 60^2)\big]\,. \end{aligned} \tag{24}$$

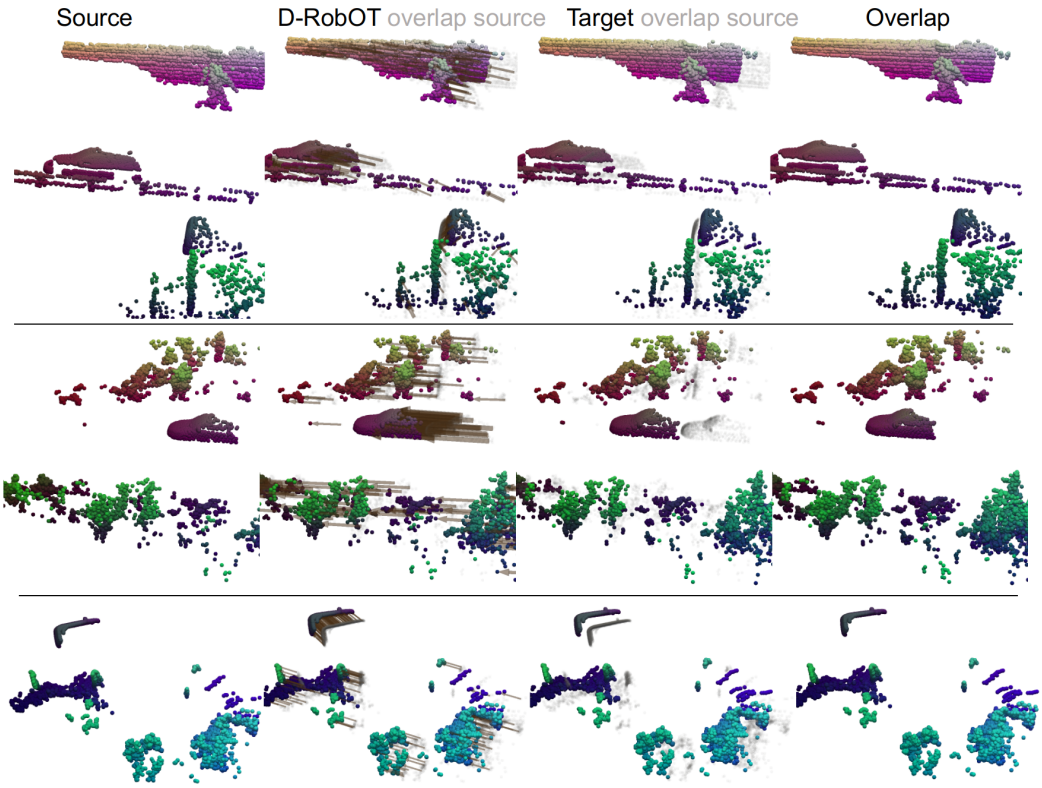


Figure 14: **Additional results for scene flow estimation on Kitti.** From left to right, the columns refer to: the source shape; the registration result for D-RobOT with a spline deformation model (the source point cloud is displayed in gray in the background, flows are displayed as brown arrows); the target shape (the source point cloud is displayed in gray in the background); and the D-RobOT result overlapped with the target. Each row corresponds to a single pair of 3D frames.

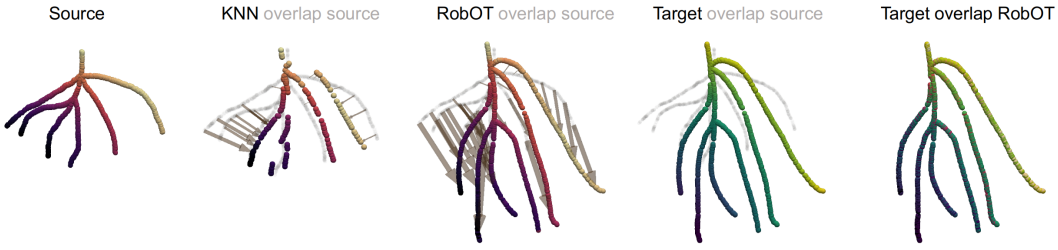


Figure 15: **RobOT matching vs Nearest Neighbor projection on a synthetic local vessel tree.** From left to right, the columns refer to: the source shape; the registration result for NN projection and RobOT followed by a local smoother (the source point cloud is displayed in gray in the background, flows are displayed as gray arrows); the target shape (the source point cloud is displayed in gray in the background); and the D-RobOT result overlapped with the target.

3. Postprocessing. To fine-tune our registration, we use the spline S-RobOT deformation of Eq. (7) with (x, y, z) coordinates as input features in $\mathbb{R}^3$. We set the *blur* parameter to $\sigma = 5$ cm and the *reach* parameter to $\tau = 80$ cm. For smoothing, we use the vessel-preserving anisotropic kernel $k(x, y)$ of Sec. A.2 with a kernel scale $s_{\text{local}} = 80$ cm.

A.5 RobOT matching vs nearest neighbor projection.

As detailed in Sec. 2.2, we use the RobOT matching of Eq. (4) as a plug-in replacement for nearest neighbor (NN) projection. At an affordable computational cost, our RobOT layer enforces a local conservation of the point density: this geometric prior is relevant for many registration tasks and mitigates accumulation artifacts that are commonly found in nearest neighbor projections [35, 40]. In accordance with the theory, we thus observe that RobOT matching outperforms nearest neighbor projection in a wide range of settings.

Local translations. When the point features p_i and q_j correspond to the (x, y, z) coordinates in $\mathbb{R}^3$, a key property of the Monge-Brenier map that is defined by Eq. (4) is that it perfectly retrieves translations and dilations [15, 35, 87]. Scene flow estimation is often close to this best case scenario for RobOT theory: after we remove points that correspond to the ground (a common pre-processing for this task), most of the displacements can be explained as local translations and small rotations of solid objects (e.g. cars or trees) in the frame of reference of the acquisition device.

On the Kitti benchmark of Fig. 4, we observe that a simple RobOT matching already performs very well for 3D scene flow estimation. Remarkably, and without any training, **vanilla RobOT matching** on high-resolution point clouds (30k points per frame) **outperforms most pre-existing deep learning methods** on medium-resolution point clouds (8,192 points per frame) in terms of speed, memory usage and accuracy. This is strong evidence that geometric methods deserve more attention from the computer vision community.

Complex structures. Going further, the mass distribution constraint of Eq. (2) is useful to register **complex shapes with appendices and branches**. This property has been discussed in depth in previous works: we refer to e.g. [36] for the matching of five-fingered hand surfaces. The (soft) constraints on the marginals of the transport plan promote a bijective matching of the fingers, preventing the thumb and the index of the source shape from being both projected on the thumb of the target.

In Fig. 15, we provide a similar example for the local registration of branching vessels. We create a synthetic pair of source and target vessels with $\mathrm{N} = 1,000$ and $\mathrm{M} = 1,200$ points respectively. We normalize their (x, y, z) coordinates to be contained in the unit cube $[0, 1]^3$ and assign uniform weights $\alpha_i = 1$ and $\beta_j = 1$ to each point – we work in a realistic unbalanced scenario. We compute the raw NN matching using a simple nearest neighbor projection from the source onto the target in $\mathbb{R}^3$. For the RobOT matching, we also use raw (x, y, z) coordinates and the squared Euclidean metric; we set the *blur* parameter to 0.0005 units in $\mathbb{R}^3$ and the *reach* parameter to 1 unit (unbalanced OT). As detailed in Suppl. A.2, we regularize both of the NN and RobOT matchings using an anisotropic muti-Gaussian-kernel with standard deviations $\{0.03, 0.05, 0.07\}$ and weights $\{0.2, 0.3, 0.5\}$.

As a post-processing step. As discussed above, RobOT matching is good at handling local translations, dilations and small free-form deformations. We propose to use it as a fast post-processing step in our D-RobOT architecture, presented in Sec. 3.2. In order to validate its utility, we perform an ablation study on the PVT1010/DirLab dataset for lung registration.

We benchmark increasingly suitable fine-tuning layers: no fine-tuning; nearest neighbor projection; RobOT matching; nearest neighbor projection with vessel-preserving smoothing; RobOT matching with vessel-preserving smoothing. Our results are detailed in Tab. 4: when used in combination with a smooth deformation model, these layers result in an increasingly higher accuracy. This confirms the main findings of our work: including **sensible geometric priors** in a modular deep learning architecture is the key to reliable state-of-the-art performance.

A.6 Computational Resources

For the evaluation of time and memory cost in Fig. 4, we compare all models on a Ubuntu server with a single 24GB NVIDIA Quadro RTX 6000 graphics processing unit (GPU) and a 10-core Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz. We trained all of our networks on a single 24GB

NVIDIA RTX 3090 GPU, except for the training of PointPWC-Net with 30k sampled points that was performed on a 48GB NVIDIA RTX A6000 GPU.

A.7 Potential for negative societal impact

Point cloud registration is a fundamental low-level task in computer vision and computer graphics. As a consequence, negative societal impact of our work may arise in a wide range of use cases. A first example is that of point cloud registration for autonomous driving: algorithm failure could lead to incorrect driving decisions. Similar concerns may arise when using these approaches to register facial scans for the purpose of person identification. In general, while our manuscript demonstrates improved performance and robustness over competing approaches, the possibility for registration failures should always be considered when applying our models in safety critical environments.