Point-TTA: Test-Time Adaptation for Point Cloud Registration Using Multitask Meta-Auxiliary Learning

Ahmed Hatem
University of Manitoba
hatema@myumanitoba.ca

Yiming Qian
University of Manitoba
yiming.qian@umanitoba.ca

Yang Wang
Concordia University
yang.wang@concordia.ca

Abstract

We present Point-TTA, a novel test-time adaptation framework for point cloud registration (PCR) that improves the generalization and the performance of registration models. While learning-based approaches have achieved impressive progress, generalization to unknown testing environments remains a major challenge due to the variations in 3D scans. Existing methods typically train a generic model and the same trained model is applied on each instance during testing. This could be sub-optimal since it is difficult for the same model to handle all the variations during testing. In this paper, we propose a test-time adaptation approach for PCR. Our model can adapt to unseen distributions at test-time without requiring any prior knowledge of the test data. Concretely, we design three self-supervised auxiliary tasks that are optimized jointly with the primary PCR task. Given a test instance, we adapt our model using these auxiliary tasks and the updated model is used to perform the inference. During training, our model is trained using a meta-auxiliary learning approach, such that the adapted model via auxiliary tasks improves the accuracy of the primary task. Experimental results demonstrate the effectiveness of our approach in improving generalization of point cloud registration and outperforming other state-of-the-art approaches.

1. Introduction

Given a pair of overlapping 3D point clouds, the goal of point cloud registration (PCR) is to estimate the 3D transformation that aligns these point clouds. PCR plays a vital role in a variety of applications, such as autonomous driving [7], augmented reality [5], and robotics [30]. Standard PCR approaches start with extracting the pointwise features. These features are matched to establish the correspondences between points in the pair. Then an outlier rejection module is used to remove outliers since the correspondences obtained by feature matching are not completely reliable. Finally, the correspondences are used to estimate the optimal 3D rotation and translation to align the two point cloud fragments. Traditional approaches establish correspondences by matching hand-crafted features and leveraging robust iterative sampling strategies such as RANSAC [17] for model estimation. However, these approaches take a long time to converge and their accuracy drops in the presence of high outliers.

To address the limitation of traditional approaches, most recent PCR methods use learning-based approaches instead of handcrafted features. These approaches first learn a model on a labeled dataset. Then the model is fixed when evaluating on unseen test data. The single set of model parameters may not be optimal for different test environments captured with different 3D scanners due to the domain shift. Our work is inspired by the success of test-time adaption (TTA) [47] in image classification, where an auxiliary task is used to update the model parameters at inference time to learn feature representation specifically for a test instance. In this paper, we introduce a TTA approach for point cloud registration that adapts to new distributions at test-time. Unlike existing 3D point cloud domain adaptation approaches [27, 54, 40, 28, 51], our method does not require any prior knowledge about the test data distributions. We adapt the model parameters in an instance-specific manner during inference and obtain a different set of network parameters for each different instance. This allows our model to better capture the uniqueness of each test instance and thus generalize better to unseen data. More importantly, our TTA formulation is a *generic* framework that can be applied in a plug-and-play manner to boost standard PCR pipelines.

Auxiliary learning has been shown to be effective to improve a predefined primary task and is used in multiple 2D computer vision tasks [35, 36, 47, 9]. Recently, there have been some attempts at using auxiliary tasks for improving the representation learning of 3D point clouds [1, 24, 44, 15]. However, using auxiliary tasks for test-time adaptation is still largely unexplored for 3D point cloud data. In this work, we introduce three self-supervised auxiliary tasks: point cloud reconstruction, feature learning, and correspondence classification. These auxiliary tasks do not require any extra supervision.

Some recent work [9] has shown that naively training the primary task and the auxiliary task together may not be optimal, since the model may be biased toward improving the auxiliary task rather than the primary task. Following [9], we use a meta auxiliary learning framework based on the model-agnostic meta learning (MAML) [16]. Each point cloud pair acts as a task in MAML. We update the model for each task using the auxiliary loss provided by the proposed three auxiliary tasks. The updated model is then used to perform the primary task. The model parameters are learned in such a way that the updated model using the auxiliary tasks improve the performance of the primary registration task. Our key contributions are summarized as follows:

- We propose a test-time adaptation approach for point cloud registration. To the best of our knowledge, this is the first work to apply test-time adaptation for 3D point cloud registration.
- We design three self-supervised auxiliary tasks to effectively extract useful features from test instances and adapt the model to unseen test distribution to improve generalization. A meta auxiliary learning paradigm is used to learn the model parameters, such that adapting the model parameter via the auxiliary tasks during testing improves the performance of the primary task.
- We perform extensive experiments on 3Dmatch [56] and KITTI [19] benchmarks to show the effectiveness of our approach in improving point cloud registration performance and achieving superior results.

2. Related Work

We review two lines of related work in point cloud registration and meta-auxiliary learning.

Point Cloud Registration. Most traditional PCR approaches consist of two modules: feature-based correspondence matching and outlier filtering.

Feature descriptors have been proposed to effectively extract the local and global features of point clouds, which are used to match correspondences in the feature space. Traditional methods use hand-crafted features such as spatial features histogram [29, 48, 18], or geometric features histogram [6, 43]. Recently, learning-based approaches have been proposed to learn 3D feature descriptors including fully convolution methods [20, 11], keypoint detection methods [3, 52, 33], and coarse-to-fine methods [55, 41].

The correspondences obtained from feature matching often include many outliers that must be filtered out for robust point cloud registration. Many traditional approaches [17, 58, 50, 8] have been proposed for robust outlier filtering of correspondences. RANSAC [17] is the most popular method, where a set of correspondences are iteratively sampled to filter outliers. RANSAC variants [45, 13, 4] have been introduced to provide new sampling strategies for fast convergence. However, these methods still have slow convergence rate and low performance in the presence of high outliers. Other methods use robust cost functions that are more effective with high outlier ratio. FGR [58] uses the Geman-McClure cost function and TEASER [50] uses the truncated least squares cost function for robust point cloud registration.

In recent years, deep learning techniques have been employed for outlier filtering. The 3D outlier filtering approaches [38, 25, 26, 12, 32, 2] follow similar ideas in 2D image matching [53, 57], where outlier filtering is defined as an inlier classification problem. 3DRegNet [38] uses the 2D correspondence selection network [53] for 3D point clouds and added a regression module for rigid transformation. DGR [12] proposes a fully convolutional network to better capture global features of correspondences and predict the inlier confidence of each correspondence. DHVR [32] leverages Hough voting in 6D transformation parameter space to identify the confidence of correspondences from Hough space to predict the final transformation. PointDSC [2] uses the spatial consistency between inlier correspondences to better prune the outliers.

Auxiliary and Meta Learning. In auxiliary learning, an auxiliary task (often self-supervised) is defined to improve the performance and generalization of a target primary task. This differs from multi-task learning where the goal is to improve performance across all tasks [35]. Auxiliary learning has been proven to be effective in multiple 2D image domain problems. [47] uses the image rotation prediction as a self-supervised auxiliary task to improve image classification. [9] uses image reconstruction as the auxiliary task to improve the primary task of deblurring. Auxiliary learning has also been studied for 3D point clouds. [44] proposes an auxiliary task that reconstruct point clouds whose parts have been randomly displaced. [24] adopts a contrastive auxiliary task [22] for better representation of 3D point clouds. Moreover, several studies have proposed self-supervised tasks [49, 1, 15] to learn domain-invariant and useful representations of point clouds from unlabelled point cloud data. [1] presents a self-supervised task of reconstructing deformations to learn the underlying structures of 3D objects. [15] introduces a pair of self-supervised tasks, including a scale prediction task and a 3D/2D projection reconstruction task to facilitate global and local features learning across different domains.

MAML [16] is a widely used meta-learning algorithm, which has been successfully employed for many 2D image domain tasks [39, 46, 35, 34, 9]. MAML learns the model parameters to fastly adapt to new tasks with few training samples and few gradient updates. [39, 46] use MAML for super-resolution problem to facilitate adaptation to unseen images. [35] proposes a meta-auxiliary framework (MAXL), in which an auxiliary label generator is trained to generate optimal labels to improve the generalization of the primary task. [9] uses a self-supervised auxiliary task in a meta learning framework for image deblurring. [34] propose to employ meta-auxiliary learning along with test-time adaptation for the problem of future depth prediction in videos.

3. Approach

Given a pair of partially overlapping 3D point clouds $X \in R^{M\times 3}$ with M points and $Y \in R^{N\times 3}$ with N points, our goal is to find an optimal 3D transformation T between the two point clouds that accurately aligns them. Our goal is to learn a model $F_\theta(X,Y) \rightarrow T$ parameterized by θ that maps (X,Y) to T. In this work, we propose a framework for point cloud registration that adapts the trained model parameters to each different input at test time, so that our model can improve generalization and performance of point cloud registration. The adaptation is achieved via self-supervised auxiliary tasks.

3.1. Auxiliary Tasks

We propose three different auxiliary tasks in our work. All these auxiliary tasks are self-supervised and do not require extra labels. So they can be used during test-time for adaptation.

Point Cloud Reconstruction. Inspired by the success of using image reconstruction as the auxiliary task [9, 36], we propose to use 3D point cloud reconstruction as one of our self-supervised auxiliary tasks. Given a point cloud P, the features of the point cloud are extracted using the feature encoder. Then, a decoder is used to reconstruct the point cloud P'. Adapting the model parameters at test time using the reconstruction auxiliary loss enables the model to take advantage of the internal features of the test instance before performing the primary task. The reconstruction loss does not require any supervision, which makes it suitable for test-time adaptation. We use $L1$ reconstruction loss as follows:

$$\ell_{rec} = ||P - P'||_1. \tag{1}$$

Self-Supervised Feature Learning. Self-supervised learning (SSL) is an active area of research. The main idea of SSL is to define some proxy self-supervised tasks to learn feature representations from data without manual annotations. We can use any existing SSL task as one of our auxiliary tasks. In our work, we adapt BYOL [22] as our self-supervised task. Different from contrastive learning, BYOL does not require negative samples. This makes it suitable for test-time adaptation. The model architecture of BYOL consists of two networks, namely the online network and the target network. Each network predicts a representation of an augmented view of the same point cloud. The idea is to train the online network to predict representations similar to the target network's predictions, so that the representations of the two augmented views are closely similar.

The online network parameterized by θ consists of a feature encoder f_θ, a feature projector z_θ and a predictor p_θ. Similarly, the target network parameterized by ξ has a feature encoder f_ξ and a feature projector z_ξ . The online network θ is trained based on the regression targets provided by the target network, while the target network ξ is the exponential moving average of the online parameters θ:

$$\xi \leftarrow \tau\xi + (1-\tau)\theta, \tag{2}$$

where $\tau \in [0,1]$ is the target decay rate.

Given a 3D point cloud P, we perform augmentation to produce two augmented versions P_v and $P_{v'}$. The point P_v is passed to the online network to obtain the projection $z_\theta = g_\theta(P_v)$ and $P_{v'}$ is passed to the target network to obtain the projection $z_\xi = g_\xi(P_{v'})$. Then we minimize the mean squared error between the normalized predictions $q_\theta(z_\theta)$ and target projections z_ξ as follows:

$$L_{\theta,\xi} = 2 - \frac{2q_\theta(z_\theta)^\top z_\xi}{\|q_\theta(z_\theta)\|^2\|\|z_\xi\|^2\|}. \tag{3}$$

We define another symmetric loss $L'_{\theta,\xi}$ by similarly passing $P_{v'}$ to the online network and P_v to the target network to compute $L'_{\theta,\xi}$. The final BYOL loss is defined as:

$$\ell_{byol} = L_{\theta,\xi} + L'_{\theta,\xi}. \tag{4}$$

Correspondence Classification. We introduce an additional self-supervised auxiliary task designed specifically for PCR. Given a 3D point cloud P, we construct an augmented point cloud P' using a randomly generated 3D transformation T by sampling a random rotation along three axes within $[0°..360°]$ and a random translation within [0cm..60cm]. The sampled transformation T is applied on each axis of point cloud to obtain P'. The feature encoder is used to extract the features of P and P'. Then these two sets of points are matched in the feature space using nearest neighbors to obtain the correspondences. Using the same outlier rejection network architecture of the primary task, this auxiliary task is trained to predict whether a correspondence is an inlier or an outlier. Since the transformation T of the point cloud is known, the ground-truth inlier correspondences C are available and the auxiliary loss does not require any manual supervision. Similar to [12, 2], the classification loss is defined as the binary cross entropy loss

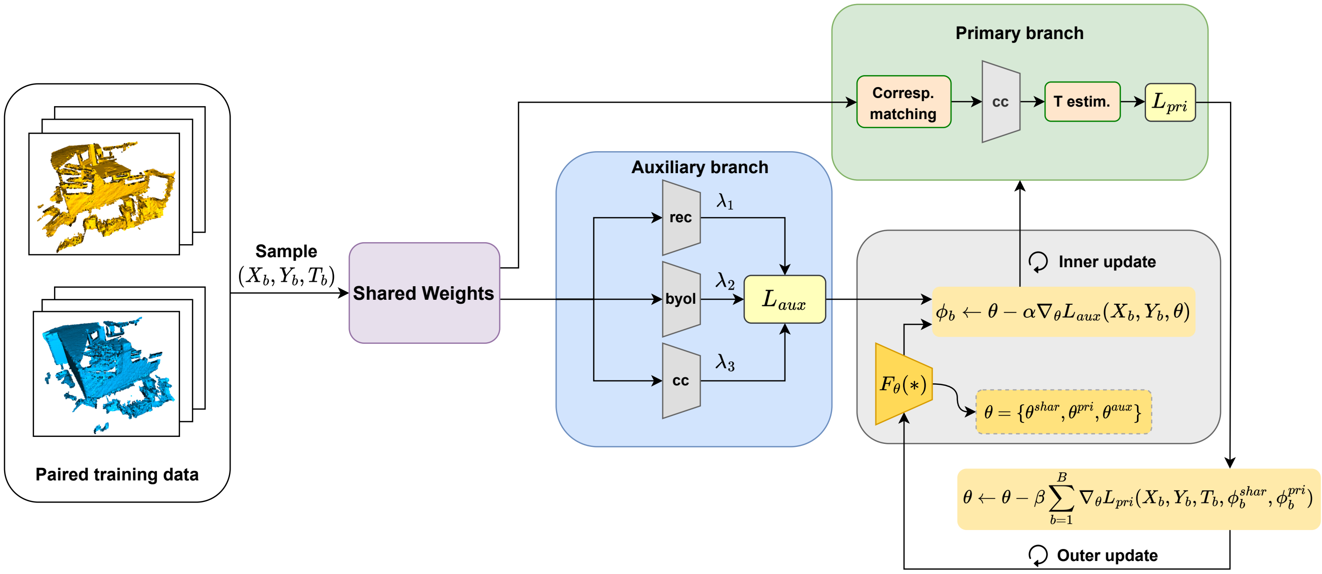
Paired training data
Sample
(X_b, Y_b, T_b)
Shared Weights
Auxiliary branch
rec
byol
cc
λ_1
λ_2
λ_3
L_{aux}
Primary branch
Corresp. matching
cc
T estim.
L_{pri}
Inner update
$\phi_b \leftarrow \theta - \alpha \nabla_\theta L_{aux}(X_b, Y_b, \theta)$
$F_\theta(*)$
$\theta = \{\theta^{shar}, \theta^{pri}, \theta^{aux}\}$
$\theta \leftarrow \theta - \beta \sum_{b=1}^{B} \nabla_\theta L_{pri}(X_b, Y_b, T_b, \phi_b^{shar}, \phi_b^{pri})$
Outer update

Figure 1: Overview of the proposed meta-auxiliary training framework. Given a pair of input point clouds during training, we first adapt the model by performing a small number of gradient updates using the auxiliary loss calculated via three auxiliary tasks including point cloud reconstruction, BYOL and correspondence classification. Then the adapted model is used to perform the primary registration task and is evaluated using the meta-objective. Finally, we update the model using the primary loss.

between the probability $p^i_{(i,j)}$ that a correspondence $C_{(i,j)}$ is an inlier and the ground-truth inliers C.

$$\ell_{cc} = \frac{1}{|M|}\left(\sum_{(i,j)\in C} \log p^i_{(i,j)} + \sum \log p^o_{(i,j)}\right), \qquad (5)$$

where $p^o = 1 - p^i$.

3.2. Model Architecture

Our model architecture consists of a shared feature encoder and two branches for the primary and auxiliary tasks. The primary branch corresponds to the point cloud registration task. The auxiliary branch corresponds to three self-supervised auxiliary tasks defined in Section 3.1. We denote the model parameters as $\theta = \{\theta^{shar}, \theta^{pri}, \theta^{aux}\}$, where θ^{shar} corresponds to the shared feature encoder, θ^{pri} is the primary branch and θ^{aux} is the auxiliary branch. Note that θ^{aux} represents the parameters of three auxiliary tasks.

Auxiliary Tasks. Our aim of the auxiliary tasks is to transfer rich and useful knowledge to improve the performance of the primary task. The overall auxiliary loss is the weighted sum of the losses for the three auxiliary tasks:

$$L_{aux} = \lambda_1 \ell_{rec} + \lambda_2 \ell_{byol} + \lambda_3 \ell_{cc}. \qquad (6)$$

Instead of fixing the values of the balancing weights λ_i ($i = 1, 2, 3$), we treat them as learnable parameters and learn their values during training. This allows the learning algorithm to automatically choose the right weights that balance the relative importance of each auxiliary task.

To train both primary and auxiliary tasks, we first follow the joint training approach in [47]. The loss of the joint training is simply the combination of the primary and auxiliary losses:

$$L_{pri}(\theta^{shar}, \theta^{pri}; X, Y, T) + L_{aux}(\theta^{shar}, \theta^{aux}; X, Y). \quad (7)$$

Note that since our auxiliary tasks are self-supervised, the auxiliary loss $L_{aux}(\cdot)$ does not need the ground-truth transformation T. To simplify the notation, we have assumed one training instance in Eq. 7. It is straightforward to generalize Eq. 7 to the entire training set by summing over all training instances.

The model learned from Eq. 7 is then used as the initialization for the meta-auxiliary learning.

Primary Task. We follow the standard learning-based network architecture for point cloud registration. First, a pair of 3D point clouds are passed to a fully convolutional network to extract the corresponding geometric pointwise features. Then the points are matched using the nearest neighbor in the feature space to obtain correspondences. These correspondences are fed to an outlier rejection network which predicts the confidence of each correspondence. Finally, given the correspondences with their associated probability weights resulting from the outlier rejection network,

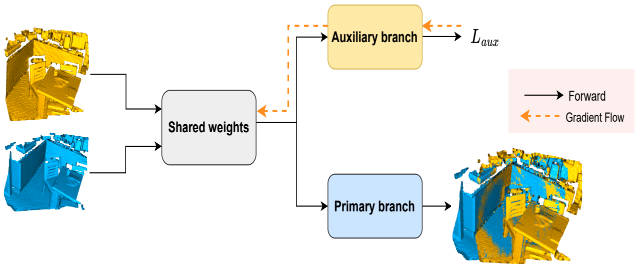

Figure 2: Overview of the proposed meta-auxiliary testing procedure. We use the auxiliary branch to fine-tune the model for each test instance using the auxiliary loss and the adapted model is used to register the input point clouds.

the weighted Procrustes approach is used to align the paired 3D scans by estimating the transformation between the two point clouds. We use $L_{pri}(\theta^{shar}, \theta^{pri}; X, Y, T)$ to denote the loss function that measures the difference between the ground-truth transformation T and the prediction $F_\theta(X, Y)$.

In this paper, we use Fully Convolutional Geometric Features (FCGF) [11] to extract pointwise features of the 3D point clouds. For the outlier rejection network, we adopt the architecture of three state-of-the-art methods including DGR [12], DHVR [32] and PointDSC [2]. However, it is importance to note that our proposed meta-auxiliary framework is agnostic to these choices and can be applied to any learning-based point cloud registration methods.

3.3. Meta-Auxiliary Learning

Our goal is to combine self-supervised auxiliary tasks along with the point cloud registration task to quickly adapt the model parameters for each test instance without the need for any extra supervision. Although jointly training the primary and auxiliary tasks will improve the generalization of our model to unknown test distribution, the updated parameters using auxiliary loss may be more biased during training to improve the auxiliary task and not the primary task. Following [9], we propose to use a meta auxiliary task scheme.

Training. We use meta-learning to train the model parameters θ to be quickly adaptable to different test distribution data, such that updating model parameters at test-time improve the primary point cloud registration task.

Given a batch of paired point clouds X_b, Y_b, and the pre-trained model parameters θ resulting from jointly training primary and auxiliary tasks on the 3DMatch dataset [56] by optimizing Eq. 7. We perform adaptation for small gradient updates using the auxiliary loss, in which all model parameters ($\phi_b^{shar}, \phi_b^{pri}, \phi_b^{aux}$) are updated:

$$\phi_b \leftarrow \theta - \alpha \nabla_\theta L_{aux}(X_b, Y_b, \theta), \tag{8}$$

where α is the adaptation learning rate. Note that since Eq. 8 is based on the auxiliary task, the adaptation can be done at test-time since it does not require the ground-truth transformation.

Algorithm 1 Meta-auxiliary training

```
Require: $X, Y, T$: training pairs with their transformation
Require: $\alpha, \beta$: learning rates
Output: $\theta$: learned parameters
Initialize the network with pre-trained weights $\theta$
while not done do
    Sample a training batch $\{X_b, Y_b, T_b\}_{b=1}^{B}$
    for each example do
        Evaluate the three auxiliary tasks:
        $L_{aux} = \lambda_1 \ell_{rec} + \lambda_2 \ell_{byol} + \lambda_3 \ell_{cc}$
        Compute adapted parameters via gradient descent:
        $\phi_b \leftarrow \theta - \alpha \nabla_\theta L_{aux}(X_b, Y_b, \theta)$
        Update auxiliary branch:
        $\theta^{aux} \leftarrow \theta^{aux} - \alpha \nabla_\theta L_{aux}(X_b, Y_b, \theta^{aux})$
    end for
    Evaluate the primary task using the adapted parameters and update:
    $\theta \leftarrow \theta - \beta \sum_{b=1}^{B} \nabla_\theta L_{pri}(X_b, Y_b, T_b, \phi_b^{shar}, \phi_b^{pri})$
end while
return $\theta$
```

Then, the adapted model($\phi_b^{shar}, \phi_b^{pri}$) will be used to perform the primary task and calculate the primary loss. This will enforce the adapted model to boost the primary task performance. The primary loss will be used to optimize the model parameters θ as:

$$\theta \leftarrow \theta - \beta \sum_{b=1}^{B} \nabla_\theta L_{pri}(X_b, Y_b, T_b, \phi_b^{shar}, \phi_b^{pri}), \tag{9}$$

where β is the meta-learning rate and B is the batch size. Note that $L_{pri}(\cdot)$ in Eq. 9 is defined in terms of the updated model ϕ_b for each instance, while the optimization is performed on the model parameters θ. The training process is summarized in Algorithm 1 and Figure 1.

Testing. During test-time, the optimized meta-learned parameters θ are adapted to a test instance that consists of a pair of 3D point clouds using the auxiliary loss as follows:

$$\phi \leftarrow \theta - \alpha \nabla_\theta L_{aux} \tag{10}$$

Then, the adapted model (ϕ^{shar}, ϕ^{pri}) is used to perform point cloud registration.

4. Experiments

We first evaluate our method on a 3D indoor dataset for the pairwise registration task. Then we analyze the generalization of our proposed model to unseen 3D outdoor datasets. Additionally, we integrate our method into

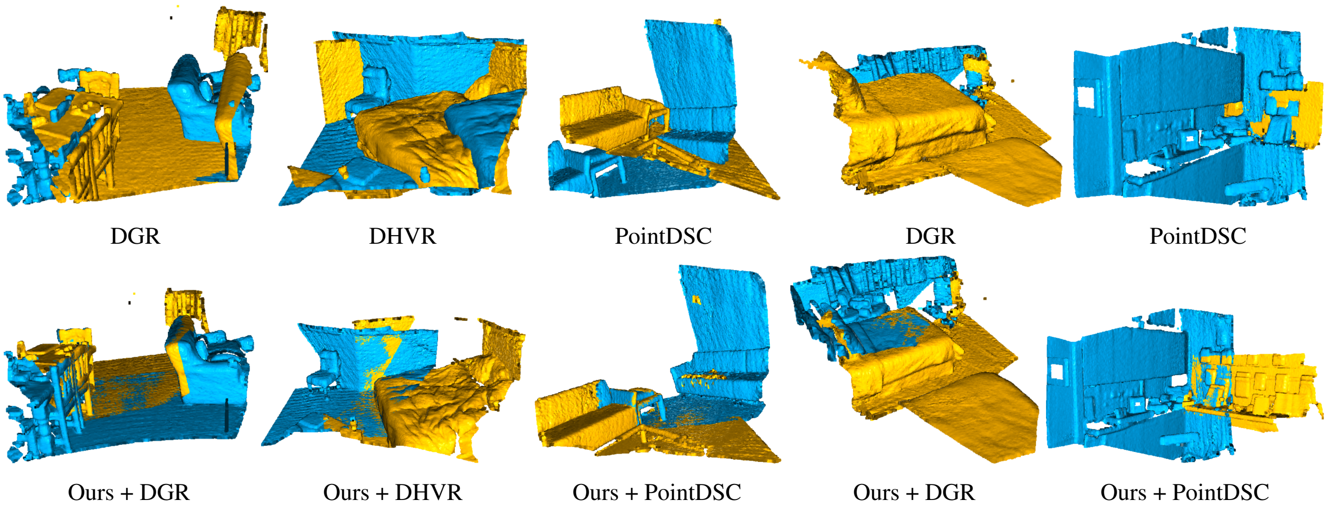

Figure 3: We propose a generic test-time adaptation framework that can be applied to boost standard point cloud registration pipeline. Here we show qualitative comparisons between the baselines and our method. The first three columns are from 3DMatch dataset [56] and the last two columns are from 3DLoMatch dataset [23]. Our proposed framework can successfully align failure examples of DGR [12], DHVR [32], and PointDSC [2].

a multi-way registration pipeline and evaluate its performance on generating final 3D reconstruction scenes. Finally, we perform extensive ablation studies to inspect each component of our approach. Additional results and ablation studies are provided in the supplementary document.

4.1. Experimental Setup

Dataset. We use the 3DMatch benchmark [56] for indoor pairwise registration. It consists of point cloud pairs with corresponding ground-truth transformations from real-world indoor scenes scanned by commodity RGB-D sensors. We follow the standard splitting strategy and evaluation protocol in 3DMatch [56], where the test data contain 1623 partially overlapping 3D point cloud scans from 8 different indoor scenes. For the outdoor dataset, we use the KITTI odometry benchmark [19] which consists of 3D outdoor scenes scanned using a Velodyne laser. We follow the train/test split in [11] to create pairwise splits, since the official benchmark does not have labels for pairwise registration. We perform voxel downsampling to generate point clouds with uniform density and set voxel size to 5cm for indoor dataset and 30cm for outdoor dataset. For multi-way registration experiment, we use the simulated Augmented ICL-NUIM dataset [10] which contains augmented indoor reconstruction scenes from RGB-D videos.

Evaluation Metrics. Following [12, 2], we report Registration Recall (RR), Rotation Error (RE) and Translation Error (TE). RE and TE are defined as:

$$RE = \arccos \frac{Tr(R^T R^*) - 1}{2}, TE = ||t - t^*||^2, \quad (11)$$

where R^* and t^* are the ground-truth rotation and translation, respectively. Registration Recall (RR) is the ratio of successful pairwise registration that its rotation error and translation error are below predefined thresholds. These thresholds are set to ($RE = 15$, $TE = 30cm$) for indoor scenes and ($RE = 5$, $TE = 60cm$) for outdoor scenes.

Implementation Details. We implement our framework in PyTorch and use the official implementation of DGR [12], DHVR [32], and PointDSC [2] as the backbones of our approach. We first jointly train primary and auxiliary tasks by optimizing the loss in Eq. 7 using the ADAM optimizer with an initial learning rate of 10^{-4} and an exponentially decayed factor of 0.99. For meta-training, the learning rates α and β are set to $2.5 \times 10e^{-5}$. We perform 5 gradient updates during training and testing to adapt the model parameters using the auxiliary loss in Eq. 6. All experiments are conducted on an NVIDIA TitanX GPU.

4.2. Main Results

We first evaluate our method on the 3DMatch dataset and report the results in Table 1. We compare our method with 5 traditional methods: FGR [58], TEASER [50], GC-RANSAC [4], RANSAC [17], CG-SAC [42], 2 unsupervised learning-based methods: LEAD [37], Ppf-foldnet

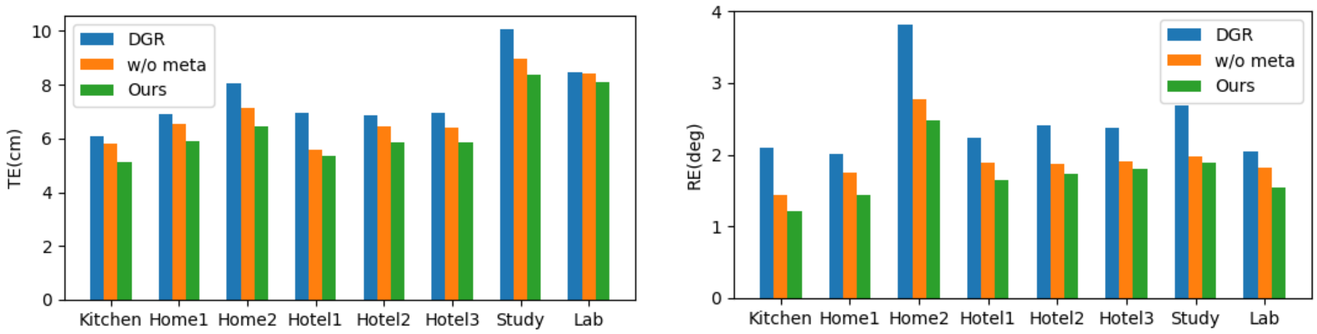

Figure 4: Registration results for different scenes on the 3DMatch dataset in terms of RE and TE. "w/o meta" refers to a variant of our method that simply optimizes Eq. 7 during training without adopting the meta-auxiliary training paradigm.

Table 1: Comparison with other state-of-the-art methods on the 3DMatch dataset [56]. ↑ (or ↓) indicates that a higher (or lower) number means better performance.

	Recall ↑	RE (deg) ↓	TE (cm) ↓
FGR [58]	78.56	2.82	8.36
TEASER [50]	85.77	2.73	8.66
GC-RANSAC [4]	92.05	2.33	7.11
RANSAC-1M [17]	88.42	3.05	9.42
RANSAC-2M [17]	90.88	2.71	8.31
RANSAC-4M [17]	91.44	2.69	8.38
CG-SAC [42]	87.52	2.42	7.66
LEAD [37]	67.15	3.39	12.01
Ppf-foldnet [14]	71.82	3.45	9.16
3DRegNet [38]	77.76	2.74	8.13
DGR [12]	91.30	2.40	7.48
Ours + DGR	**92.45**	**1.71**	**6.39**
DHVR [32]	91.40	2.08	6.61
Ours + DHVR	**92.28**	**1.75**	**6.42**
PointDSC [2]	92.85	2.08	6.51
PointDSC-reported	93.28	2.06	6.55
Ours + PointDSC	**93.47**	**1.70**	**6.21**

[14], and 4 supervised learning-based methods: 3DRegNet [38], DGR [12], DHVR [32], PointDSC [2]. All learning-based methods are trained on the 3DMatch dataset and follow the same experiment setup for fair comparison. As shown in Table 1, our method improves the registration recall of DGR [12] and DHVR [32] by about 1%, and PointDSC [2] by about 0.5%, as well as the RE and TE have significantly decreased for all three backbones (on average 7.3% and 21%). More importantly, our method with PointDSC [2] as backbone outperforms all other state-of-the-art methods. Figure 3 shows qualitative results comparison on challenging examples of 3DMatch [56] and 3DLoMatch [23] datasets when applying our TTA method to DGR [12]. Our meta-auxiliary framework enables the model to better capture the internal features of each test-instance, leading to performance improvement.

Table 2: Results of cross-dataset generalization. Here we train the model on 3DMatch [56] and evaluate on KITTI [19]. The results of training on KITTI and evaluating on 3DMatch can be found in the supplementary document.

	Recall ↑	RE (deg) ↓	TE (cm) ↓
DGR	95.24	0.44	23.25
Ours + DGR	**97.36**	**0.34**	**21.16**
DHVR	95.82	0.39	22.17
Ours + DHVR	**98.01**	**0.32**	**21.18**
PointDSC	97.15	0.36	21.74
Ours + PointDSC	**98.23**	**0.33**	**20.86**

Figure 4 shows the 3DMatch registration results per scene. As an ablation study, we remove the meta learning diagram and simply optimize the joint loss Eq. 7 during training[1]. By doing so, we obtain superior results than the backbone DGR [12], which demonstrates that our proposed auxiliary tasks can complement the primary PCR task and thus improves accuracy. Moreover, our final meta-auxiliary framework achieves the best. Figure 5 shows the robustness of our approach under different rotation and translation error thresholds.

4.3. Ablation and Additional Results

We perform additional experiments and ablation studies to further analyze our proposed method.

Outdoor Registration Generalization. Although our method achieves the best performance over all the traditional and learning-based methods, the main advantage of our method is the ability to generalize to unseen test distribution. In order to evaluate the generalization of our method, we perform a cross-dataset experiment on both 3DMatch [56] and KITTI [19] datasets, where the trained model on 3DMatch [56] is used to test on KITTI [19] and vice versa. As shown in Table 2, our method shows a significant improvement on all evaluation metrics when

[1] At test-time, we still perform Eq. 10 for TTA.

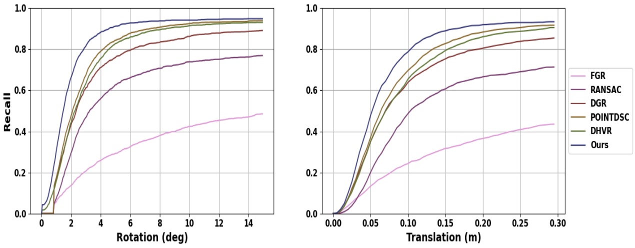

Figure 5: Comparison of the registration recall on the 3DMatch dataset between our approach and other state-of-the-art methods by varying the translation error and the rotation error thresholds. Our framework with PointDSC as backbone outperforms all other methods for all thresholds.

Table 3: Registration results on ETH dataset [21]. We report the registration recall per scene as well as the average recall across all scenes.

	Gazebo		Wood		AVG
	Summer	Winter	Summer	Autumn	
DGR	92.63	83.28	79.86	71.34	81.78
Ours + DGR	**95.28**	**87.95**	**81.73**	**82.45**	**86.85**
DHVR	93.78	82.74	81.07	75.32	83.22
Ours + DHVR	**95.96**	**87.13**	**82.25**	**80.79**	**86.53**
PointDSC	94.21	89.68	83.14	78.42	86.36
Ours + PointDSC	**94.89**	**91.49**	**87.02**	**85.22**	**89.65**

training on 3DMatch dataset [56] and evaluating on KITTI dataset [19], demonstrating the effectiveness of the proposed framework. To further inspect the generalization of our method, we test our framework on another widely-used outdoor dataset, namely ETH dataset [21]. In Table 3, we report the registration results of our approach when evaluating on ETH dataset [21]. We can observe that our method improves the performance of the baselines across all scenes with a good margin. In particular, the average recall of DGR [12] is significantly improved by about 5%, which sheds light on the generalization capability of our approach.

Robustness to Low-Overlapping Point Clouds. To further validate the robustness of our method, we evaluate our method on a dataset with low-overlapping ratio between input point clouds, namely 3DLoMatch [23]. This dataset is constructed from the 3DMatch benchmark [56] and has a low-overlapping ratio (10%-30%) between 3D point cloud fragments. Figure 6 compares the inlier ratio between 3DLoMatch [23] and 3DMatch [56], which shows that 3DLoMatch [23] is more challenging due to the lower inlier ratio. We use the model trained on 3DMatch [56] for evaluation and report our results in Table 4. Our approach outperforms all other methods, demonstrating the robustness of our approach to low-overlapping scenarios. More importantly, this validates the robustness of our approach to the percentage of template and target overlaps where 3Dmatch [56] contains overlapping ratios (≥30%) and 3DLoMatch [23] contains low-overlapping ratios (10%-30%). The evaluation results on both datasets show the superiority of our approach among the baselines under different ratios.

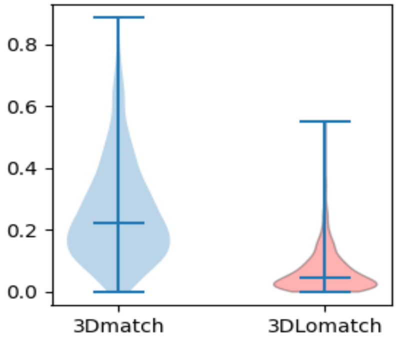

Figure 6: Comparison between the distribution of the inlier ratio of correspondences obtained by feature matching on 3DMatch [56] and 3DLoMatch [23] benchmarks. The registration task is more challenging with a lower inlier ratio.

Table 4: Robustness to low-overlapping point clouds on the 3DLoMatch dataset [23] with low-overlapping ratio between 3D point cloud segments. We train on 3DMatch and evaluate on 3DLoMatch.

	Recall ↑	RE (deg) ↓	TE (cm) ↓
DGR	43.80	4.17	10.82
Ours + DGR	**50.73**	**4.06**	**10.52**
DHVR	54.46	4.13	10.54
Ours + DHVR	**57.32**	**3.83**	**10.26**
PointDSC	56.10	3.87	10.39
Ours + PointDSC	**57.81**	**3.79**	**10.15**

Table 5: Multiway registeration results on Augmented ICL-NUIM dataset evaluated by ATE(cm) where lower is better.

	Living1	Living2	Office1	Office2	AVG
FGR	78.97	24.91	14.96	21.05	34.98
RANSAC	110.9	19.33	14.42	17.31	40.49
DGR	21.06	21.88	15.76	11.56	17.57
Ours + DGR	**18.32**	**16.12**	**12.24**	**10.44**	**14.28**
DHVR	22.91	16.37	12.58	10.90	15.69
Ours + DHVR	**18.46**	**13.59**	**12.43**	**9.56**	**13.51**
PointDSC	20.25	15.58	13.56	11.30	15.18
Ours + PointDSC	**15.73**	**12.07**	**12.15**	**9.78**	**12.43**

Multiway Registration for 3D Reconstruction. Point cloud registration is a critical step for various 3D applications. In this section, we present the effect of pairwise registration performance on obtaining more accurate and robust 3D reconstruction scenes. Following [12, 2], we integrate our method into a 3D reconstruction pipeline[10]. Given RGB-D scans, 3D fragments are generated from the scene. Next, we perform pairwise registration using our

Table 6: Ablation studies on framework components: Auxiliary Learning, Meta Learning, and Test-time Adaptation.

	Recall ↑	RE (deg) ↓	TE (cm) ↓
DGR	91.31	2.40	7.48
DGR + Aux.	91.42	2.25	7.06
DGR + TTA (w/o meta)	91.86	1.88	6.54
DGR + Meta-Aux. (w/o TTA)	92.28	1.71	6.40
DGR + full framework	**92.45**	**1.71**	**6.39**

Table 7: Ablation studies on the three auxiliary tasks: Point Cloud Reconstruction (rec), Correspondence Classification (cc), and Feature Learning (byol).

	Recall ↑	RE (deg) ↓	TE (cm) ↓
DGR [12]	91.31	2.43	7.34
DGR + rec	92.24	1.71	6.42
DGR + (rec, cc)	92.38	**1.69**	6.40
DGR + (rec, cc, byol)	**92.45**	1.71	**6.39**

method to align all fragments. Finally, multi-way registration [10] is used to optimize the fragment poses using pose graph optimization [31]. We use the model trained on 3DMatch to further demonstrate the generalization of our method and evaluate our approach on Augmented ICL dataset using Absolute Trajectory Error (ATE). As shown in Table 5, our method achieves the lowest error compared to all other methods.

Methodology Components. To study the effectiveness of the proposed framework, we conduct ablation experiments on 3DMatch dataset [56] and evaluate the effect of each component of the proposed framework. We consider DGR [12] as the backbone in our experiments and report the results after applying each component to DGR [12]. Specifically, we compare the results between our method's three major components: Auxiliary Learning, Meta Learning, Test-time Adaptation. We first investigate the effect of our proposed Auxiliary Learning method by jointly training the primary and three auxiliary tasks by optimizing the loss in Eq. 7. This shows the strength of the proposed auxiliary tasks acting as a regularizer during training. Then, we study the impact of combining test-time adaptation with auxiliary learning, in which the auxiliary tasks are used to update the model parameters at test-time by optimizing the auxiliary loss in Eq. 6. Furthermore, we show the effect of the proposed meta-auxiliary learning paradigm elaborated in Algorithm 1 in learning optimal model parameters. However, we fixed the model parameters at test-time. Finally, we report the results of our final framework integrating Auxiliary Learning, Meta Learning, and Test-time Adaptation.

As reported in Table 6, auxiliary learning improves the registration results across all evaluation metrics compared to DGR [12]. This demonstrates that the proposed auxiliary tasks can complement the registration task, leading to performance improvement. Combining auxiliary learning with TTA has further improved the performance of registration recall by 0.44%, and decreased the TE and RE by 0.52cm and 0.37 deg, respectively. As TTA allows the auxiliary tasks to transfer useful features of test-instance to the primary registration task, it enhances the registration performance. Moreover, the proposed meta-auxiliary training method greatly boosts the performance, which demonstrates the effectiveness of training tasks using meta-learning terminology such that the meta-objective enforces the auxiliary tasks to improve the primary task performance. Finally, our final framework further boosts the registration performance by fine-tuning the model parameters at test-time.

Analysis of Auxiliary Tasks. We conduct an additional ablation study on the 3DMatch dataset [56] to investigate the importance of each auxiliary task in our approach in improving the registration performance. As shown in Table 7, the auxiliary reconstruction task significantly boosts the registration recall of DGR [12] by 0.92%. Also, Translation Error (TE) and Rotation Error (RE) greatly drop by 13% and 30%, respectively. These evaluation metrics are further improved when combining the auxiliary correspondence classification task to the reconstruction task. This demonstrates the impact of multiple auxiliary tasks in transferring additional features to the primary task and enhancing registration results. Finally, our final three auxiliary tasks achieve a higher registration recall of 92.45% and a lower Translation Error (TE) of 6.39cm. However, the Rotation Error (RE) was slightly worse when compared to the two auxiliary tasks results.

5. Conclusion

We have introduced a novel test-time adaptation framework for point cloud registration using multitask meta-auxiliary learning. Previous work usually follows a supervised learning approach to train a model on a labeled dataset and fix the model during evaluation on unseen test data. In contrast, our framework is designed to effectively adapt the model parameters at test time for each test instance to boost the performance. We have introduced three self-supervised auxiliary tasks to improve the the primary registration task. Furthermore, we have used a meta-auxiliary learning paradigm to train the primary and auxiliary tasks, so that the adapted model using auxiliary tasks improve the performance of the primary task. Extensive experiments show the effectiveness of the proposed approach in improving the registration performance and outperforming state-of-the-art methods.

References

[1] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point clouds. *IEEE Winter Conference on Applications of Computer Vision*, 2021. 1, 2

[2] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3, 5, 6, 7, 8

[3] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[4] Daniel Barath and Jiří Matas. Graph-cut ransac. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 6, 7

[5] Mark Billinghurst, Adrian J. Clark, and Gun A. Lee. A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*, 8:73–272, 2015. 1

[6] Hui Chen and Bir Bhanu. 3d free-form object recognition in range images using local surface patches. *IEEE International Conference on Pattern Recognition*, 2004. 2

[7] Siheng Chen, Baoan Liu, Chen Feng, Carlos Vallespi-Gonzalez, and Carl Wellington. 3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception. *IEEE Signal Processing Magazine*, 2020. 1

[8] Zhi Chen, Kun Sun, Fan Yang, and Wenbing Tao. Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration. *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[9] Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jingshan Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 3, 5

[10] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 6, 8, 9

[11] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. *IEEE International Conference on Computer Vision*, 2019. 2, 5, 6

[12] Christopher Bongsoo Choy, Wei Dong, and Vladlen Koltun. Deep global registration. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 3, 5, 6, 7, 8, 9

[13] Ondřej Chum, Jiri Matas, and Josef Kittler. Locally optimized ransac. *Symposium of the German Association for Pattern Recognition*, 2003. 2

[14] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. *European Conference on Computer Vision*, 2018. 7

[15] Hehe Fan, Xiaojun Chang, Wanyue Zhang, Yi Cheng, Ying Sun, and Mohan S. Kankanhalli. Self-supervised global-local structure modeling for point cloud domain adaptation with reliable voted pseudo labels. *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2

[16] Chelsea Finn, P. Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning*, 2017. 2

[17] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1, 2, 6, 7

[18] Andrea Frome, Daniel F. Huber, Ravi Krishna Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. *European Conference on Computer Vision*, 2004. 2

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 2013. 2, 6, 7, 8

[20] Zan Gojcic, Caifa Zhou, Jan Dirk Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[21] Zan Gojcic, Caifa Zhou, Jan Dirk Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 8

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 2020. 2, 3

[23] Shengyu Huang, Zan Gojcic, Mikhail (Misha) Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 6, 7, 8

[24] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. *IEEE International Conference on Computer Vision*, 2021. 1, 2

[25] Xiaoshui Huang, Wentao Qu, Yifan Zuo, Yuming Fang, and Xiaowei Zhao. Gmf: General multimodal fusion framework for correspondence outlier rejection. *IEEE Robotics and Automation Letters*, 2022. 2

[26] Xiaoshui Huang, Yangfu Wang, Sheng Li, Guofeng Mei, Zongyi Xu, Yucheng Wang, Jian Zhang, and Mohammed Bennamoun. Robust real-world point cloud registration by inlier detection. *Computer Vision and Image Understanding*, 2022. 2

[27] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Perez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[28] Peng Jiang and Srikanth Saripalli. Lidarnet: A boundary-aware domain adaptation model for point cloud semantic segmentation. *IEEE International Conference on Robotics and Automation*, 2021. 1

[29] Andrew Edie Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 1999. 2

[30] Ioannis Kostavelis and Antonios Gasteratos. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 2015. 1

[31] Rainer Kümmerle, Giorgio Grisetti, Hauke Malte Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. *IEEE International Conference on Robotics and Automation*, 2011. 9

[32] Junha Lee, Seungwook Kim, Minsu Cho, and Jaesik Park. Deep hough voting for robust global registration. *IEEE International Conference on Computer Vision*, 2021. 2, 5, 6, 7

[33] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. *IEEE International Conference on Computer Vision*, 2019. 2

[34] Huan Liu, Zhixiang Chi, Yuanhao Yu, Yang Wang, Jun Chen, and Jin Tang. Meta-auxiliary learning for future depth prediction in videos. *IEEE Winter Conference on Applications of Computer Vision*, 2023. 3

[35] Shikun Liu, Andrew Davison, and Edward Johns. Selfsupervised generalization with meta auxiliary learning. *Advances in Neural Information Processing Systems*, 2019. 1, 2, 3

[36] Kaiyue Lu, Nick Barnes, Saeed Anwar, and Liang Zheng. From depth what can you see? depth completion via auxiliary image reconstruction. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 3

[37] Marlon Marcon, Riccardo Spezialetti, Samuele Salti, Luciano Silva, and Luigi Di Stefano. Unsupervised learning of local equivariant descriptors for point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 6, 7

[38] G. Dias Pais, Pedro Miraldo, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C. Nascimento, and Rama Chellappa. 3dregnet: A deep neural network for 3d point registration. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 7

[39] Seobin Park, Jinsu Yoo, Donghyeon Cho, and Jiwon Kim andTae Hyun Kim. Fast adaptation to super-resolution networks via meta-learning. *European Conference on Computer Vision*, 2020. 3

[40] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 2019. 1

[41] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kaiping Xu. Geometric transformer for fast and robust point cloud registration. *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[42] Siwen Quan and Jiaqi Yang. Compatibility-guided sampling consensus for 3-d point cloud registration. *IEEE Transactions on Geoscience and Remote Sensing*, 2020. 6, 7

[43] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. *IEEE International Conference on Robotics and Automation*, 2009. 2

[44] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 2019. 1, 2

[45] Ruwen Schnabel, Roland Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 2007. 2

[46] Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. Meta-transfer learning for zero-shot super-resolution. *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[47] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. *International Conference on Machine Learning*, 2020. 1, 2, 4

[48] Federico Tombari, Samuele Salti, and Luigi di Stefano. Unique shape context for 3d data description. *3D Object Retrieval*, 2010. 2

[49] Aoran Xiao, Jiaxing Huang, Dayan Guan, Xiaoqin Zhang, Shijian Lu, and Ling Shao. Unsupervised point cloud representation learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[50] Heng Yang, J. Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 2021. 2, 6, 7

[51] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1

[52] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. *European Conference on Computer Vision*, 2018. 2

[53] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal V. Fua. Learning to find good correspondences. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[54] Li Yi, Boqing Gong, and Thomas A. Funkhouser. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1

[55] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust point cloud registration. *Advances in Neural Information Processing Systems*, 2021. 2

[56] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas A. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 5, 6, 7, 8, 9

[57] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. *IEEE International Conference on Computer Vision*, 2019. 2

[58] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. *European Conference on Computer Vision*, 2016. 2, 6, 7