Optimization over Disentangled Encoding: Unsupervised Cross-Domain Point Cloud Completion via Occlusion Factor Manipulation

Jingyu Gong[1⋆], Fengqi Liu[1⋆], Jiachen Xu[1], Min Wang[2], Xin Tan[3], Zhizhong Zhang[3], Ran Yi[1], Haichuan Song[3], Yuan Xie[3⋆⋆], and Lizhuang Ma[1,3,4⋆⋆]

[1] Shanghai Jiao Tong University, Shanghai, China
[2] SenseTime Research, Shanghai, China
[3] East China Normal University, Shanghai, China
[4] Qing Yuan Research Institute, SJTU, Shanghai, China
{gongjingyu,liufengqi,xujiachen,tanxin2017,ranyi}@sjtu.edu.cn, wangmin@sensetime.com, {zzzhang,hcsong}@cs.ecnu.edu.cn, yxie@cs.ecnu.edu.cn, ma-lz@cs.sjtu.edu.cn

Abstract. Recently, studies considering domain gaps in shape completion attracted more attention, due to the undesirable performance of supervised methods on real scans. They only noticed the gap in input scans, but ignored the gap in output prediction, which is specific for completion. In this paper, we disentangle partial scans into three (domain, shape, and occlusion) factors to handle the output gap in cross-domain completion. For factor learning, we design view-point prediction and domain classification tasks in a self-supervised manner and bring a factor permutation consistency regularization to ensure factor independence. Thus, scans can be completed by simply manipulating occlusion factors while preserving domain and shape information. To further adapt to instances in the target domain, we introduce an optimization stage to maximize the consistency between completed shapes and input scans. Extensive experiments on real scans and synthetic datasets show that ours outperforms previous methods by a large margin and is encouraging for the following works. Code is available at https://github.com/azuki-miho/OptDE.

Keywords: Point cloud completion · Cross-domain · Disentanglement

1 Introduction

Shape completion in which we infer complete shapes given partial ones, attracted lots of attention recently, for its wide application in robotics, path planning and VR/AR [6,14,42]. However, real scan completion is quite challenging due to the irregularity of point cloud and absence of complete real shapes for training.

Previous methods [30,15,35] had widely exploited completion on virtual 3D models like ShapeNet [5] and achieved desirable performance. That is attributed

⋆ Equal Contribution.
⋆⋆ Corresponding Author.

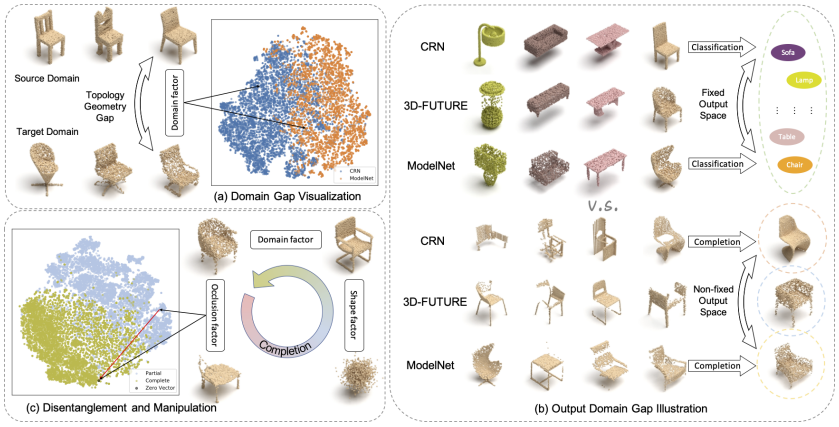

Fig. 1. (a) presents the domain gap between objects from the same category but different datasets where topology and geometry patterns vary a lot, as well as the feature distribution. (b) illustrates the output domain gap which is specific for completion task in contrast to cross-domain classification. In (c), we disentangle any partial scan into three independent factors, and completion can be simply implemented by setting occlusion factors to zero vector (red arrow) while well preserve shape and domain features.

to the availability of complete shapes, and corresponding partial point clouds can be obtained through virtual scanning [41]. However, it is hard to use supervised methods for real point cloud completion, because complete shapes of real objects are usually unavailable for supervision. Meanwhile, completion networks trained on virtual 3D model are commonly not able to generalize well to infer complete real objects, especially when there are large domain gaps between the synthesized shapes and real objects [6]. So, the key for recovering real scans is to handle the cross-domain completion, where geometry and topology of objects from the same category are different between various datasets as shown in Figure 1 (a).

To alleviating the influence of domain gaps, pcl2pcl [6] trained two auto-encoders and an adaptation network to transform the latent code of a real scan to that of complete virtual one for each category. Then, the decoder for complete virtual shapes can map the transformed codes into complete shapes. However, they ignored a serious problem which is specific for completion task. In contrast to cross-domain shape classification whose output space (*i.e.*, categories) is invariant to different input domains, the predicted complete point cloud should correspond to the domain of input partial shapes (see Figure 1 (b)). *That means an output domain gap in completion task, as illustrated by the domain factor distribution of complete shapes from CRN [32] and ModelNet [38] after t-SNE in Figure 1 (a).* Therefore, decoder trained to infer complete virtual shape cannot complete real scans. Recently, based on TreeGAN[27] and GAN Inversion [22], ShapeInversion [42] fine-tuned the decoder trained on virtual shapes during optimization stage for better performance. The underlying principle is that, fine-tuning according to real scans could adapt the decoder to real instances, allevi-

ating output domain gap to some extent. However, adaptation in optimization alone is not enough, especially when the output domain gaps are quite large.

To handle the output domain gap, we assume the category of input partial shapes is known in advance as previous works [6,33,42], and introduce an intensively disentangled representation for partial shapes of each category via comprehensive consideration of both shape completion and output domain gap. Specifically, in our assumption, there are three generative factors, *i.e.*, occlusion, domain and domain-invariant shape factor, underlying a given partial cloud as shown in Figure 1 (c). For complete point clouds from the same category but different domains, they usually share the same semantic parts but quite different topology and geometry patterns. Thus, we attempt to disentangle any complete shape into a domain factor and a domain-invariant shape factor. While for a given complete shape, the partial point clouds generated from scanning vary a lot due to the occlusion caused by different scanning view-point [41]. So, we also introduce an occlusion factor which indicates the view-point for partial scans.

Based on this assumption, any partial scan can be disentangled into these factors no matter it is virtual or real, and we also assume the occlusion factor will be all-zero if the input shape is complete. Thus, point cloud completion can be simply implemented by manipulating the occlusion factor (setting to zero vector, see the occlusion factor manipulation in Figure 1 (c)), while shape and domain information can be well-preserved in the output prediction.

To thoroughly explore these three factors, we design several components to facilitate the disentanglement. (a) It is noteworthy that occlusion in partial shapes is usually caused by scanning at fixed view-point. Therefore, we introduce a self-supervised view-point prediction task for better occlusion factor/feature learning. (b) We take a domain discriminator judging whether a shape is virtual or real to extract domain factor, and employ another domain discriminator to decouple the domain information from the shape feature in an adversarial way. Additionally, we utilize the completion task to ensure the domain and domain-invariant shape factors are enough to infer complete shapes. (c) Inspired by PMP [10], for more intensive disentanglement, we derive a new factor permutation consistency regularization by randomly swapping the factors between samples, and introduce an inverse structure of auto-encoder, decoder-encoder for swapped factor reconstruction to ensure these factors independent to each other.

To further adapt our prediction to each partial instance in the target domain, we embrace a collaboration of regression and optimization. We use the trained encoder to obtain the disentangled factors of input cloud and set the occlusion factor to zero. Then, the combined factors can give a good initial prediction given the decoder. Later, Chamfer Distance between partial input and masked prediction [42] is used to fine-tune these factors and decoder within several iterations. Thanks to the collaboration, our method can give prediction 100$\times$ faster than pure optimization method like [42] and achieve much better performance.

To evaluate the performance on cross-domain completion, we test on real datasets ScanNet [8], MatterPort3D [4] and KITTI [12] like previous works [6,42]. We also utilize two additional point cloud completion datasets 3D-FUTURE [9]

and ModelNet [38] with complete shapes for more comprehensive evaluation. The experimental results demonstrate our method can well cover the gap in output prediction and significantly improve cross-domain completion performance.

2 Related Works

Point Cloud Completion. Inspired by PointNet [25], PCN [41] designed an auto-encoder with folding operation [40] for shape completion. Later, a great performance boost has been brought in virtual shape completion, where paired shapes are available for training [30,34,20,39,43,23,35]. To generalize to real scans, pcl2pcl [6] trained two auto-encoders for virtual complete shapes and real partial scans, and an adaptation network to map the latent codes of real scans to that of virtual complete shapes. Cycle4Completion [33] added a reverse mapping function to maintain shape consistency. Meanwhile, ShapeInversion [42] searched for the latent code that best reconstruct the shapes during the optimization stage, given a generator for complete virtual shapes. Even though they fine-tuned the generator for better adaptation of partial scans, it is far from enough when the domain gaps are large.

Compared with these methods, we attempt to handle the domain gaps in the output prediction, which is specific for generation or completion task. In our method, we disentangle occlusion factor and domain factor for better completion of real scans while preserving domain-specific patterns.

Disentangled Representation Learning. Disentangled factor learning was explored under the concept of "discovering factorial codes" [2] by minimizing the predictability of one factor given remaining units [26]. Based on VAE [17], FactorVAE proposed to disentangle the latent code to be factorial and independent [16]. Factors can be more easily used for image manipulation and translation after disentanglement [19,21]. To reduce the domain discrepancy, domain-specific and domain-invariant features were disentangled for better recognition performance [37,24]. To ensure the disentanglement, additional constraints on factor through recombination were used in motion and 3D shape modeling [1,7].

Inspired by their works, we design a disentangled space consisting of occlusion, domain and domain-invariant shape factors for cross-domain shape completion where complete shapes in target domain are unavailable for training.

Collaboration of Regression and Optimization. Methods based on optimization and regression correspond to generative and discriminative models respectively, and recently their cooperation became prevailing for its speed and performance. In pose estimation, optimization was used on a good initial pose estimated by a regression approach [28]. Later, results obtained by optimization further supervised the regression network [18]. To speed up the process of finding latent code z of the generator that best reconstruct an image in image manipulation, an encoder is used for good initialization before further optimization [3].

Inspired by these methods, we further optimize the prediction given by disentangled encoding within several iterations to adapt to each input partial instance in the target domain, making it 100× faster than optimization-based completion method and achieve much better performance.

3 Method

Overview. The framework of our two-stage method is shown in Figure 2. In the first stage, we attempt to disentangle the input partial point cloud (unpaired source domain partial shapes generated from complete ones through real-time rendering and target domain partial scans) into three factors, naming view-based occlusion factor, domain factor, and domain-invariant shape factor as shown in Figure 2 (a). Here, we design a view-point prediction task in a self-supervised manner to disentangle the occlusion information caused by scanning. Concurrently, domain discriminators are taken to disentangle the domain-specific information from domain-invariant shape features. Later, three factors will be combined to output reconstructed partial point clouds. Meanwhile, we can simply predict the complete shapes by setting the occlusion factor to zero vector. Additionally, the independence of these three factors are ensured by randomly permuting the factors within a batch and keeping combined factors consistent after a decoder-encoder structure as presented by Figure 2 (b). In the second stage, for completion of specific partial point clouds, we optimize the disentangled factors and decoder obtained from stage 1 within several iterations to better adapt to each input partial point cloud instance (Figure 2 (c)).

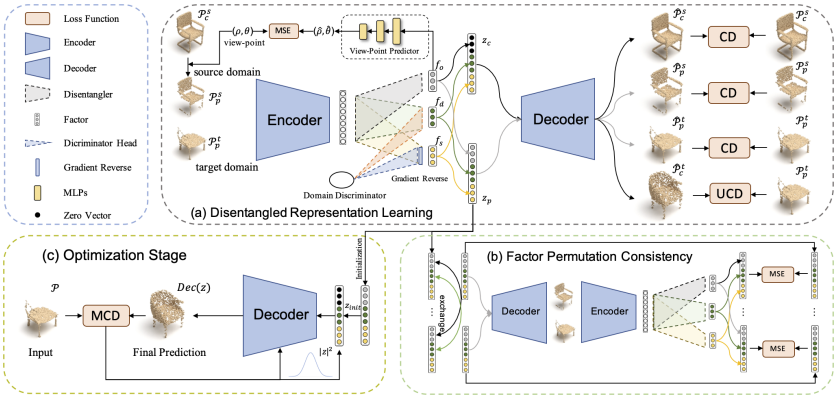

Fig. 2. Overall framework of Optimization over Disentangled Encoding. (a) shows the supervision given by view-point prediction, domain discrimination, reconstruction and completion. (b) shows the procedure of factor permutation consistency where factors are more intensively disentangled. Here, the Encoder, Decoder and Disentanglers are shared with (a). (c) shows the optimization procedure over disentangled factors of completed partial shapes and the Decoder is initialized by pre-trained model from (a).

3.1 Disentangled Representation for Completion

According to the disentanglement assumption [13,10], there are intrinsic factors $\{f_i\}_{i=1}^{l}$ in much lower dimension that generate the observed samples in high dimension point cloud space $\mathcal{P}(f_1, \cdots, f_l)$. In our method, we attempt to disentangle the common partial point cloud into three independent factors, including the occlusion factor f_o, domain factor f_d, and domain-invariant shape factor f_s.

View-based Occlusion Factor. Partial shapes are mainly caused by occlusion, since a complete shape will generate various partial clouds when scanned from different view-points. So, scanning view-point plays a key role in point cloud completion and we aim to disentangle the view-based occlusion factor specially [31].

1) View-point Prediction: To disentangle the view-based occlusion factor, we design a view-point prediction task in a self-supervised manner. Here, we assume the view-point is located in a unit sphere. As shown in Figure 2 (a), we first randomly generate azimuth and elevation angles (ρ, θ) as the view-point direction, and rotate the complete point cloud accordingly. Then, based on z-buffer [29], we design a real-time implementation to render the complete shapes in a fast non-differentiable way and obtain the partial clouds in source domain.

Then, the generated partial shapes will be fed into the shared encoder to extract common features, and a specific disentangler is utilized to obtain occlusion factor f_o. For better factor learning, we introduce a view-point predictor module VP, consisting of several MLPs, to predict the view-point of point cloud $(\hat{\rho}, \hat{\theta}) = VP(f_o)$. The loss for view-point prediction can be formulated as follows:

$$\mathcal{L}_{vp} = (\rho - \hat{\rho})^2 + (\theta - \hat{\theta})^2. \tag{1}$$

Here, we choose to predict the azimuth and elevation angle directly rather than the rotation matrix due to their independence and simplicity.

2) Occlusion Factor Manipulation: Additionally, we assume it is the occlusion factor that makes the point cloud incomplete, and the decoder will predict the complete shape when the occlusion factor is zeroed out. So, for the same input partial point cloud, it can generate reconstructed partial shapes and complete objects by simply manipulating the occlusion factor. The corresponding latent factors for reconstructed partial shapes (z_p) and complete ones (z_c) are:

$$z_p = f_o \otimes f_d \otimes f_s, \quad z_c = \mathbf{0} \otimes f_d \otimes f_s, \tag{2}$$

where $\otimes$ indicates vector concatenation. Therefore, the reconstructed partial shape and completed shape are $\hat{\mathcal{P}}_p = Dec(z_p)$ and $\hat{\mathcal{P}}_c = Dec(z_c)$, respectively, where $Dec(\cdot)$ is the decoder. Chamfer Distance (CD) or Unidirectional Chamfer Distance (UCD) between the output prediction and target shapes are used to supervise the whole network. We take the form of CD as previous works [42,23]:

$$\mathcal{CD}(\mathcal{P}_1, \mathcal{P}_2) = \frac{1}{|\mathcal{P}_1|} \sum_{p_1 \in \mathcal{P}_1} \min_{p_2 \in \mathcal{P}_2} \|p_1 - p_2\|_2^2 + \frac{1}{|\mathcal{P}_2|} \sum_{p_2 \in \mathcal{P}_2} \min_{p_1 \in \mathcal{P}_1} \|p_1 - p_2\|_2^2, \tag{3}$$

and UCD is formulated as follows:

$$\mathcal{UCD}(\mathcal{P}_1, \mathcal{P}_2) = \frac{1}{|\mathcal{P}_1|} \sum_{p_1 \in \mathcal{P}_1} \min_{p_2 \in \mathcal{P}_2} \|p_1 - p_2\|_2^2. \tag{4}$$

Here, we use $\mathcal{CD}(\cdot)$ to supervise the reconstructed partial shape from source $\hat{\mathcal{P}}_p^s$ and target domains $\hat{\mathcal{P}}_p^t$, and predicted complete shape from source domains $\hat{\mathcal{P}}_c^s$. For inferred complete shape in target domain $\hat{\mathcal{P}}_c^t$ where GT are not available, we take UCD for guidance. Therefore, the loss function for reconstruction and completion can be expressed by

$$\mathcal{L}_{rec}^s = \mathcal{CD}(\mathcal{P}_p^s, \hat{\mathcal{P}}_p^s), \quad \mathcal{L}_{rec}^t = \mathcal{CD}(\mathcal{P}_p^t, \hat{\mathcal{P}}_p^t), \tag{5}$$

and

$$\mathcal{L}_{com}^s = \mathcal{CD}(\mathcal{P}_c^s, \hat{\mathcal{P}}_c^s), \quad \mathcal{L}_{com}^t = \mathcal{UCD}(\mathcal{P}_p^t, \hat{\mathcal{P}}_c^t), \tag{6}$$

where $\mathcal{P}_p^s$ and $\mathcal{P}_c^s$ are input partial cloud and corresponding complete shape Ground Truth (GT) from source domain respectively, and $\mathcal{P}_p^t$ indicates the input partial shape from the target domain.

Domain Factor. To keep the domain-specific features of input partial shapes well preserved in the output prediction, we extract the domain factor to provide domain clues for the decoder. As shown in Figure 2 (a), we utilize a specific disentangler to extract domain factors from common hidden features and introduce a domain discriminator to guide the learning of domain information. Here, our network will predict whether an input partial shape comes from the source domain or target domain according to the domain factor. Then, the domain labels, which are generated automatically, will supervise the domain prediction through cross-entropy loss, guiding the learning of domain factor.

Domain-invariant Shape Factor. To make the shape factor domain-invariant, we also utilize the domain discriminator to distinguish the shape factor. However, a gradient reverse layer [11] is utilized between the domain discriminator and shape factor, where gradient will be reversed during back-propagation. Thus, the shape factor can learn to be domain-invariant in an adversarial way. To learn the shape information, this factor will be combined with domain factor to predict the complete point cloud, and the output prediction will be supervised in Eq. (6).

3.2 Factor Permutation Consistency

The independence of each disentangled factor is pursued for an intensive disentanglement [26,10]. To satisfy this property, we introduce a factor permutation consistency loss for the disentanglement of partial point cloud. Specifically, we first feed a batch of B samples into encoder to extract common features, then three separate disentanglers are used to extract occlusion factors $\{f_o^i\}_{i=1}^B$, domain factors $\{f_d^i\}_{i=1}^B$ and domain-invariant shape features $\{f_s^i\}_{i=1}^B$ respectively.

In order to make the shape factors invariant to different domain and occlusion situations, we choose to generate random permutations of occlusion features f_o^i or domain features f_d^i to form new combinations of factors:

$$\tilde{z}^i = f_o^j \otimes f_d^i \otimes f_s^i \quad \text{or} \quad \tilde{z}^i = f_o^i \otimes f_d^j \otimes f_s^i, \tag{7}$$

where j is a permutation of i. In our implementation, we attempt to permute occlusion factors or domain factors alternately. As we need to make sure the extracted factors are independent to the remaining factors, an inverse structure of auto-encoder, saying decoder-encoder as shown in Figure 2 (b), is designed to keep factor permutation consistency with the following loss:

$$\mathcal{L}_{cons} = \sum_{i}^{B} \| Enc(Dec(\tilde{z}^i)) - \tilde{z}^i \|_2^2, \tag{8}$$

where Enc consists of the shared encoder and disentanglers, and Dec indicates the decoder. It is noteworthy that we only add factor permutation consistency loss halfway, when the three factors have been learned preliminarily in encoder.

3.3 Optimization over Disentangled Encoding

Based on the well-trained disentangled representation, we can obtain a complete version of partial point cloud by simply manipulating the occlusion features. To make the overlapping parts between prediction and input partial shape instance look more similar, we introduce a collaboration of regression and optimization method to fine-tune latent factors and decoders within only a few iterations.

Given the pre-trained auto-encoder from Figure 2 (a)-(b) ($Enc^\dagger$,$Dec^\dagger$) and partial point cloud $\mathcal{P}$, we first obtain the disentangled factors:

$$f_o \otimes f_d \otimes f_s = Enc^\dagger(\mathcal{P}), \tag{9}$$

and obtain the initial latent factors of complete shape through:

$$z_{init} = \mathbf{0} \otimes f_d \otimes f_s, \tag{10}$$

as shown in Figure 2 (c). Meanwhile, the pre-trained decoder is utilized to initialize the output predictor $Dec_{init} = Dec^\dagger$. Then, we attempt to optimize the disentangled factors and decoder together to better adapt to input partial point clouds by optimizing the following function $\mathcal{L}_{op}$:

$$z^*, Dec^* = \arg\min_{z, Dec} \mathcal{L}_{op}(Dec(z), \mathcal{P}, z), \tag{11}$$

and the final prediction can be expressed by $Dec^*(z^*)$.

To construct the loss function, for all points of $\mathcal{P}$, we first find their k nearest neighbors in $Dec(z)$ and the union of all neighboring points form the masked point cloud $M(Dec(z))$ like ShapeInversion [42]. Then, Chamfer Distance between the partial point cloud and masked complete shape $\mathcal{MCD}(\mathcal{P}_1, \mathcal{P}_2) =$

$\mathcal{CD}(M(\mathcal{P}_1), \mathcal{P}_2)$ is used to maximize the similarity. Meanwhile, we also take a regularization of latent factors. All in all, the optimization target is:

$$\mathcal{L}_{op}(Dec(z), \mathcal{P}, z) = \mathcal{CD}(M(Dec(z)), \mathcal{P}) + \|z\|_2^2. \tag{12}$$

Compared with pure optimization method [42], our optimization stage can converge much faster and give much better predictions, since the disentangled factors and well pre-trained decoder have already covered the domain gaps in prediction and give much better initialization for further instance-level adaptation which can be evidenced obviously in Sec 4.4.

4 Experiments

To show the effectiveness of our method and demonstrate our statement, we treat CRN [32] as our source domain and evaluate the proposed method on the target domain including real-world scans from ScanNet [8], MatterPort3D [4] and KITTI [12] as well as synthesized shape completion dataset 3D-FUTURE [9] and ModelNet [38]. Following previous works [6,42], we assume the category of partial clouds are known in advance and train a separate model for each category.

4.1 Datasets

CRN. We take CRN derived from ShapeNet [5] as our source domain. It provides 30,174 partial-complete pairs from eight categories where both partial and complete shapes contain 2,048 points. Here, we take 26,863 samples from six shared categories between CRN and other datasets for training and evaluation.

Real-World Scans. Similar to previous works [6,42], we evaluate the performance of our method on partial point cloud from real scans. There are three sources for real scans, saying ScanNet, MatterPort3D, and KITTI. The tables and chairs in ScanNet and MatterPort3D, and cars in KITTI are used for performance evaluation. We re-sample the input scans to 2,048 points for unpaired training and inference to match the virtual dataset.

3D-FUTURE. To evaluate the performance on more realistic shapes, we generate another point cloud completion dataset from 3D-FUTURE [9]. The models in 3D-FUTURE are much more close to real objects. Similarly, we obtain partial shapes and complete ones with 2,048 points from 5 different view-points. Because 3D-FUTURE only contains indoor furniture models, we only take five shared categories of furniture for point cloud completion.

ModelNet. We generate a shape completion dataset ModelNet using models from ModelNet40 [38]. We synthesize the partial shape through virtually scanning and generate complete ones by randomly sampling points in the surface like previous works [41,31]. 2,048 points are taken for both partial and complete shapes to match the CRN dataset. In order to test the adaptation ability, we take the shared categories of ModelNet40 and CRN for evaluation.

4.2 Implementation

All experiments can be conducted on a machine with GTX 1080Ti and 64GB RAM. Here, we take PointNet [25] as our encoder to extract common features with dimension $1,024$. Then, we take three separate disentanglers consisting of two MLPs to extract $f_o \in \mathbb{R}^{96}$, $f_d \in \mathbb{R}^{96}$, and $f_s \in \mathbb{R}^{96}$, and a TreeGCN [27] as our decoder. More details is available at https://github.com/azuki-miho/OptDE.

4.3 Metrics and Results

Metrics. For real scans without ground truth, we use Unidirectional Chamfer Distance (UCD) and Unidirectional Hausdorff Distance (UHD) from the partial input to the predicted complete shapes as our metric following previous works [6,36,42]. For more comprehensive evaluation of our completion performance on cross-domain datasets, we take mean Chamfer Distance as our metric for the brand new datasets 3D-FUTURE and ModelNet like previous works [41,30] where complete shapes are available for testing.

Here, we first compare our method with the prevailing unsupervised cross-domain completion methods on real-world datasets of ScanNet, MatterPort3D and KITTI. The results are reported in Table 1 where UCD and UHD are used as metrics for evaluation. In this table, DE indicates regression method only using disentangled encoding shown in Figure 2 (a)-(b), and OptDE shows the results of optimization over disentangled encoding (Figure 2 (c)). As shown, disentangled encoding significantly improves the completion performance on real-world scans, and optimization over the disentangled encoding can further refine the results according to the input partial shapes. That is because our method can cover the domain gaps in output prediction between different datasets and adapt to various instances even within the target domain. We also show the qualitative results in Figure 3 where our predictions correspond well to input partial scans.

Methods	ScanNet		MatterPort3D		KITTI
	Chair	Table	Chair	Table	Car
pcl2pcl [6]	17.3/10.1	9.1/11.8	15.9/10.5	6.0/11.8	9.2/14.1
ShapeInversion[42]	3.2/10.1	3.3/11.9	3.6/10.0	3.1/11.8	2.9/13.8
+UHD [42]	4.0/9.3	6.6/11.0	4.5/9.5	5.7/10.7	5.3/12.5
Cycle4Compl. [33]	5.1/6.4	3.6/5.9	8.0/8.4	4.2/6.8	3.3/5.8
DE(Ours)	2.8/**5.4**	2.5/5.2	3.8/6.1	2.5/5.4	1.8/**3.5**
OptDE(Ours)	**2.6**/5.5	**1.9**/**4.6**	**3.0**/**5.5**	**1.9**/**5.3**	**1.6**/**3.5**

Table 1. Cross-domain completion results on real scans. We take [UCD ↓ / UHD ↓] as our metrics to evaluate the performance, and the scale factors are 10^4 for UCD and 10^2 for UHD. +UHD indicates UHD loss is used during training.

Additionally, we report the completion results of our method and previous works on target domain 3D-FUTURE in Table 2, and only complete shapes of

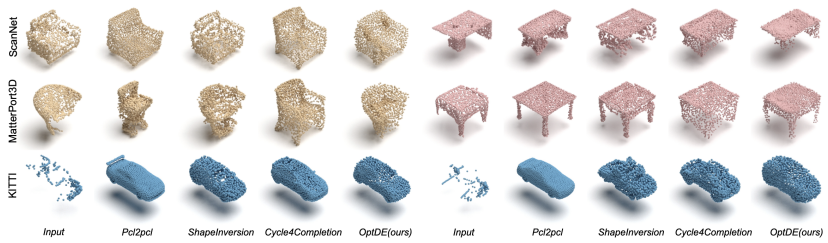

Fig. 3. Visualization results on the data of ScanNet, MatterPort3D and KITTI. Partial point clouds, predictions of pcl2pcl, ShapeInversion, Cycle4Completion and our methods are presented separately from the left to the right.

CRN and partial point clouds of 3D-FUTURE are used for training for fair comparison. In this dataset, our method significantly outperforms other competitors. Again, collaboration of regression and optimization can improve the performance by adapting to each instance. Figure 4 (a) gives the visualization results of our method and shows the qualitative improvement over previous works.

Methods	Cabinet	Chair	Lamp	Sofa	Table	Avg.
Pcl2pcl [6]	57.23	43.91	157.86	63.23	141.92	92.83
ShapeInversion [42]	38.54	26.30	48.57	44.02	108.60	53.21
Cycle4Compl. [33]	32.62	34.08	77.19	43.05	40.00	45.39
DE(Ours)	28.62	22.18	30.85	38.01	27.43	29.42
OptDE(Ours)	**28.37**	**21.87**	**29.92**	**37.98**	**26.81**	**28.99**

Table 2. Results of cross-domain completion on 3D-FUTURE. We evaluate the performance of each method using [CD↓] and scale-up factor is 10^4.

We further compare the cross-domain completion performance on target domain dataset ModelNet. The results of cross-domain completion on this dataset are reported in Table 3 where disentangled encoding alone can outperform previous methods by a large margin. In addition, optimization over the disentangled encoding can further boost the performance especially for hard categories.

Moreover, we provide the qualitative results of different methods in Figure 4 (b). As can be seen, our method can well adapt to input partial shapes from different domains.

4.4 Ablation Study

In this section, we will conduct more experiments to evaluate the effectiveness of our proposed method from different aspects and prove our claims. Without

Methods	Plane	Car	Chair	Lamp	Sofa	Table	Avg.
Pcl2pcl [6]	18.53	17.54	43.58	126.80	38.78	163.62	68.14
ShapeInversion [42]	3.78	15.66	22.25	60.42	22.25	125.31	41.61
Cycle4Compl. [33]	5.77	11.85	26.67	83.34	22.82	21.47	28.65
DE(Ours)	2.19	**9.80**	15.11	42.94	21.45	10.26	16.96
OptDE(Ours)	**2.18**	**9.80**	**14.71**	**39.74**	**19.43**	**9.75**	**15.94**

Table 3. Results of cross-domain completion on ModelNet. We take [CD↓] as our metric to evaluate the performance of each method which has been scaled by 10^4.

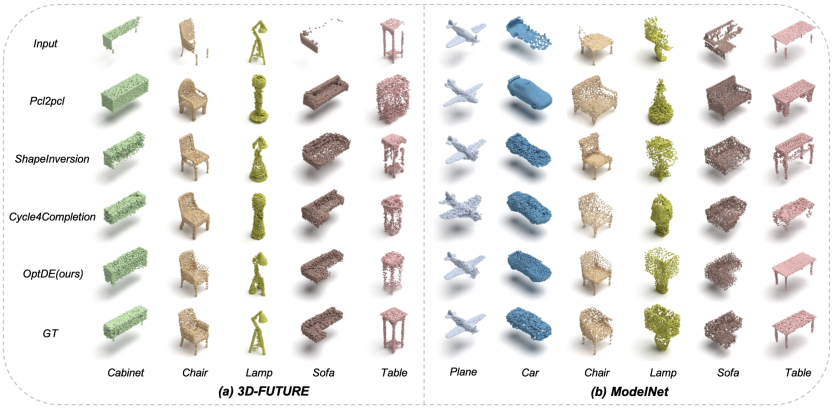

Fig. 4. Visualization results on the test set of 3D-FUTURE and ModelNet. The images from the top to bottom are input partial clouds, results given by pcl2pcl, ShapeInversion, Cycle4Completion and ours, and Ground Truth respectively.

loss of generality, we mainly utilize CRN as the source domain and evaluate on the target domain ModelNet.

Optimization over Disentangled Encoding. In order to evaluate the effectiveness of different parts in our method and test how far away from a perfect cross-domain completion method, we conduct ablation studies as follows. We first train the network with the same structure using only paired point clouds from CRN and evaluate the performance on ModelNet which is taken as our baseline. Then, we add Disentangled Representation Learning (Figure 2 (a)), Factor Permutation Consistency, and Optimization stage gradually. Additionally, we evaluate the best performance that can be brought by our backbones through training using paired data from both source domain CRN and target domain ModelNet, which is usually named as the oracle. We report all the results in Table 4.

It shows that our method can greatly handle the domain gaps in the output space and well preserve domain-specific patterns in predictions thanks to the disentanglement of occlusion factor and domain factor. Permutation consistency loss and optimization over the disentangled representation can both boost the

Methods	Plane	Car	Chair	Lamp	Sofa	Table	Avg.
Baseline	5.41	10.05	22.82	67.25	22.44	53.14	30.19
DE w/o Consistency	2.27	10.05	15.36	46.18	22.08	11.09	17.84
+ Consistency	2.19	9.80	15.11	42.94	21.45	10.26	16.96
+ Optimization	2.18	9.80	14.71	**39.74**	19.43	9.75	15.94
Oracle	**1.51**	**6.58**	**10.52**	41.98	**9.94**	**7.87**	**13.07**

Table 4. Ablation study of occlusion factor supervision on ModelNet. [CD↓]($\times 10^4$) is taken as our metric to evaluate the performance improvement and distance to oracle.

performance. Even though, the improvement on car and sofa category is minor and that is because the samples in source domain have covered most samples in the target domains but the distribution is quite different. Compared with the oracle, there are still gaps to be bridged. Thus, this paper may inspire more work to focus on how to transfer the knowledge of virtual shapes to real objects given only virtual complete shapes and real partial scans.

Occlusion Factor Manipulation. In order to show the learning of disentangled occlusion factor and prove our claims, we take four original partial point clouds $\{\mathcal{P}_i\}_{i=1}^4$ that are scanned from different view-points, and then utilize the shared encoder and disentanglers to obtain the occlusion factors, domain factors and domain-invariant shape factors. After that, we replace the occlusion factors of $\mathcal{P}_1$ and $\mathcal{P}_3$ by those of $\mathcal{P}_2$ and $\mathcal{P}_4$, and obtain new generated point clouds through the decoder as shown in Figure 5.

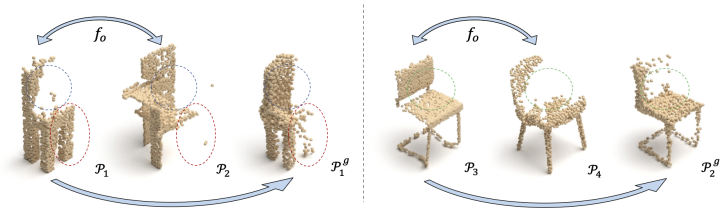

Fig. 5. Visualization of occlusion factor manipulation. The disentangled occlusion factors of $\mathcal{P}_1$ and $\mathcal{P}_3$ are replaced by the occlusion factors of $\mathcal{P}_2$ and $\mathcal{P}_4$. The new latent factors can generate brand-new partial point clouds through the pre-trained decoder.

We can see the back and seat of $\mathcal{P}_1$ is occluded (blue circle), and the right front leg (red circle) of $\mathcal{P}_2$ is occluded. After replacing the occlusion factor, the right front leg of $\mathcal{P}_1^g$ is occluded due to the occlusion factor while shape and domain information is well preserved. We can also see the occlusion factor manipulation effect in $\mathcal{P}_2^g$. This indicates a disentangled representation can provide a much easier way to control the occlusion through simple factor manipulation.

Initialization in Optimization. Our method pursue a collaboration of regression and optimization where disentangled encoding provides a good initialization for optimization. Here, we provide two representative examples of optimization progression in Figure 6 where our method can provide a desirable prediction, and the optimization can converge within about 4 iterations thanks to this good initialization of latent code and decoder. Compared with ours, ShapeInversion converges much slower and may even converge to a sub-optimal solution.

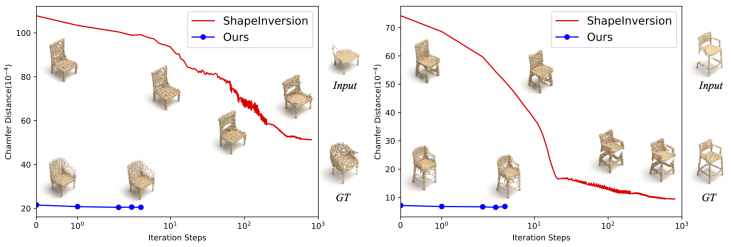

Fig. 6. Optimization progression on ModelNet test set. Compared with ShapeInversion, our method can converge 100× faster (0.12s for 4 iterations v.s. 23.56s for 800 iterations on a single GTX 1080Ti) and easily circumvent sub-optimal solution.

5 Conclusion

In this paper, we propose the very first method OptDE to deal with the output domain gap in shape completion. We introduce a disentangled representation consisting of three essential factors for any partial shape, and shape completion can be implemented by simply manipulating the occlusion factor while preserving shape and domain features. To further adapt to each partial instance in the target domain, we introduce a collaboration of regression and optimization to ensure the consistency between completed shapes and input scans. For comprehensive evaluation on cross-domain completion, we treat CRN as the source domain and evaluate on real-world scans in ScanNet, MatterPort3D and KITTI as well as synthesized datasets 3D-FUTURE and ModelNet. Results show that our method outperforms previous methods by a large margin which may inspire more works to focus on cross-domain point cloud completion.

Limitation&Discussion. Since all previous methods assume the category of partial shapes to be known and trained in category-specific way, we believe it will be better to train a unified model for cross-domain completion of all categories.

Acknowledgments. This work is sponsored by the National Key Research and Development Program of China (No. 2019YFC1521104), the National Natural Science Foundation of China (No. 61972157,72192821), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Sailing Program (22YF1420300), Shanghai Science and Technology Commission (21511101200) and SenseTime Collaborative Research Grant.

References

1. Aberman, K., Li, P., Lischinski, D., Sorkine-Hornung, O., Cohen-Or, D., Chen, B.: Skeleton-aware networks for deep motion retargeting. ACM Transactions on Graphics (TOG) **39**(4), 62–1 (2020)
2. Barlow, H.B., Kaushal, T.P., Mitchison, G.J.: Finding minimum entropy codes. Neural Computation **1**(3), 412–423 (1989)
3. Bau, D., Strobelt, H., Peebles, W., Wulff, J., Zhou, B., Zhu, J.Y., Torralba, A.: Semantic photo manipulation with a generative image prior. In: SIGGRAPH (2020)
4. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. In: 2017 International Conference on 3D Vision (3DV). pp. 667–676. IEEE Computer Society (2017)
5. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
6. Chen, X., Chen, B., Mitra, N.J.: Unpaired point cloud completion on real scans using adversarial training. In: International Conference on Learning Representations (2020)
7. Cosmo, L., Norelli, A., Halimi, O., Kimmel, R., Rodola, E.: Limp: Learning latent shape representations with metric preservation priors. In: European Conference on Computer Vision. pp. 19–35. Springer (2020)
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
9. Fu, H., Jia, R., Gao, L., Gong, M., Zhao, B., Maybank, S., Tao, D.: 3d-future: 3d furniture shape with texture. arXiv preprint arXiv:2009.09633 (2020)
10. Fumero, M., Cosmo, L., Melzi, S., Rodolà, E.: Learning disentangled representations via product manifold projection. In: (ICML) (2021)
11. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. pp. 1180–1189. PMLR (2015)
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)
13. Gonzalez-Garcia, A., van de Weijer, J., Bengio, Y.: Image-to-image translation for cross-domain disentanglement. In: NeurIPS (2018)
14. Hou, J., Dai, A., Nießner, M.: Revealnet: Seeing behind objects in rgb-d scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2098–2107 (2020)
15. Huang, Z., Yu, Y., Xu, J., Ni, F., Le, X.: Pf-net: Point fractal network for 3d point cloud completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7662–7670 (2020)
16. Kim, H., Mnih, A.: Disentangling by factorising. In: International Conference on Machine Learning (ICML). pp. 2649–2658. PMLR (2018)
17. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: (ICLR) (2014)
18. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2252–2261 (2019)
19. Liu, A.H., Liu, Y.C., Yeh, Y.Y., Wang, Y.C.F.: A unified feature disentangler for multi-domain image translation and manipulation. In: Proceedings of the 32nd

International Conference on Neural Information Processing Systems. p. 2595–2604 (2018)
20. Liu, M., Sheng, L., Yang, S., Shao, J., Hu, S.M.: Morphing and sampling network for dense point cloud completion. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 11596–11603 (2020)
21. Liu, Y.C., Yeh, Y.Y., Fu, T.C., Wang, S.D., Chiu, W.C., Wang, Y.C.F.: Detach and adapt: Learning cross-domain disentangled deep representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8867–8876 (2018)
22. Ma, F., Ayaz, U., Karaman, S.: Invertibility of convolutional generative networks from partial measurements. Advances in Neural Information Processing Systems **31** (2018)
23. Pan, L., Chen, X., Cai, Z., Zhang, J., Zhao, H., Yi, S., Liu, Z.: Variational relational point completion network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8524–8533 (2021)
24. Peng, X., Huang, Z., Sun, X., Saenko, K.: Domain agnostic learning with disentangled representations. In: International Conference on Machine Learning. pp. 5102–5112. PMLR (2019)
25. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 652–660 (2017)
26. Schmidhuber, J.: Learning factorial codes by predictability minimization. Neural computation **4**(6), 863–879 (1992)
27. Shu, D.W., Park, S.W., Kwon, J.: 3d point cloud generative adversarial network based on tree structured graph convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3859–3868 (2019)
28. Sigal, L., Balan, A., Black, M.: Combined discriminative and generative articulated pose and non-rigid shape estimation. Advances in neural information processing systems **20**, 1337–1344 (2007)
29. Straßer, W.: Schnelle kurven-und flächendarstellung auf grafischen sichtgeräten. Ph.D. thesis (1974)
30. Tchapmi, L.P., Kosaraju, V., Rezatofighi, H., Reid, I., Savarese, S.: Topnet: Structural point cloud decoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 383–392 (2019)
31. Wang, H., Liu, Q., Yue, X., Lasenby, J., Kusner, M.J.: Unsupervised point cloud pre-training via occlusion completion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9782–9792 (2021)
32. Wang, X., Ang Jr, M.H., Lee, G.H.: Cascaded refinement network for point cloud completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 790–799 (2020)
33. Wen, X., Han, Z., Cao, Y.P., Wan, P., Zheng, W., Liu, Y.S.: Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13080–13089 (2021)
34. Wen, X., Li, T., Han, Z., Liu, Y.S.: Point cloud completion by skip-attention network with hierarchical folding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1939–1948 (2020)
35. Wen, X., Xiang, P., Han, Z., Cao, Y.P., Wan, P., Zheng, W., Liu, Y.S.: Pmp-net: Point cloud completion by learning multi-step point moving paths. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7443–7452 (2021)

36. Wu, R., Chen, X., Zhuang, Y., Chen, B.: Multimodal shape completion via conditional generative adversarial networks. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16. pp. 281–296. Springer (2020)
37. Wu, X., Huang, H., Patel, V.M., He, R., Sun, Z.: Disentangled variational representation for heterogeneous face recognition. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 9005–9012 (2019)
38. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
39. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: Grnet: Gridding residual network for dense point cloud completion. In: European Conference on Computer Vision. pp. 365–381. Springer (2020)
40. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 206–215 (2018)
41. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 2018 International Conference on 3D Vision (3DV). pp. 728–737. IEEE (2018)
42. Zhang, J., Chen, X., Cai, Z., Pan, L., Zhao, H., Yi, S., Yeo, C.K., Dai, B., Loy, C.C.: Unsupervised 3d shape completion through gan inversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1768–1777 (2021)
43. Zhang, W., Yan, Q., Xiao, C.: Detail preserved point cloud completion via separated feature aggregation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16. pp. 512–528. Springer (2020)