SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

February 2026, Vol. 69, Iss. 2, 122106:1–122106:18
https://doi.org/10.1007/s11432-024-4543-3

Memory-based diverse-category single-view 3D reconstruction

Haoyu GUO[1,2*], Ying LI[1,2] & Chunyan DENG[1,2]

[1]*College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[2]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China*

Received 8 October 2024/Revised 1 February 2025/Accepted 23 March 2025/Published online 23 October 2025

Abstract Single-view object reconstruction aims to recover the geometric structure from a single image, which has wide applications in 3D modeling and virtual reality. The existing methods are limited to complex annotations or single-category models, which affect their generalizability and practical applications. To tackle this problem, we propose a memory-based single-view reconstruction network called M-SRN. Given a collection of images, M-SRN can generate high-fidelity reconstructions across diverse categories. Our main contributions here are three approaches to leverage memory representations. First, a foreground perceptron module was developed through memory-representation-based contrastive learning, enabling M-SRN to reconstruct raw image collections. Second, a purified memory-based cross-category feature compensation module was proposed to enhance dataset-level instance consistency. Finally, a dynamic neighbor consistency enhancement module based on intra-class memory prototypes was proposed to mitigate the inherent ambiguity of single-view supervision. M-SRN was validated using synthetic and real-world datasets. Experiments demonstrate that M-SRN outperforms state-of-the-art weakly supervised methods and achieves results comparable to 2D-supervised and 3D-supervised methods.

Keywords single-view reconstruction, neural networks, mesh generation, neural rendering, contrastive learning

Citation Guo H Y, Li Y, Deng C Y. Memory-based diverse-category single-view 3D reconstruction. Sci China Inf Sci, 2026, 69(2): 122106, https://doi.org/10.1007/s11432-024-4543-3

1 Introduction

Three-dimensional (3D) shape reconstruction serves as a fundamental task that involves capturing the shapes and appearances of real objects. Traditional methods [1,2] typically require multiple viewpoints or specialized equipment, which can be costly and complex. Single-view 3D reconstruction (SVR) aims to infer the 3D shape of an object from a single image and has attracted considerable attention. Owing to its simplicity, it has various applications such as augmented reality [3] and robotics [4].

Limited by a single viewpoint, early SVR methods [5–7] used 3D structures corresponding to input images as labels to train the model for a limited category. They used an encoder-decoder network structure to reconstruct the point clouds, meshes, or voxels. Although these methods perform well on synthetic datasets, their performance on real-world datasets is unsatisfactory. To enhance the generalization capability, researchers are gradually shifting towards exploring weakly supervised SVR methods and diverse-category SVR methods.

Weakly supervised SVR methods focus on reducing the annotations in category-specific reconstructions. Initially, some methods [8–10] used multi-view images and associated pose information as annotations instead of 3D data. Furthermore, some approaches [11–13] eliminated the need for multi-view or pose annotations. Recent methods [14,15] attempted to use only single-view masks for supervision, thereby making the training process more flexible. Unicorn [16] overcame the need for any form of supervision to achieve unsupervised single-category SVR. Diverse-category SVR methods use a single model to reconstruct images from multiple categories. Some methods [17,18] used 3D Gaussian point clouds as representation to achieve diverse-category reconstruction. ZeroShape [19] has further broken through category limitations to achieve zero-shot reconstruction.

However, these methods are limited to single categories or complex annotations. Some methods [20,21] have attempted to achieve diverse category reconstruction under weakly supervised conditions. ShapeClipper [21] utilizes a consistency enhancement module to achieve diverse category SDF reconstruction, requiring only mask annotation corresponding to the input image. However, their consistency reconstruction relies on fixed images for enhancement

* Corresponding author (email: ghy1999812@outlook.com)

and focuses on intra-category instances, thus limiting the reconstruction accuracy. In addition, their method depends on labor-intensive manual mask annotations. Although offline models can extract masks, inaccuracies in some instances can lead to failed reconstructions.

To address the above issues, a memory-based single-view reconstruction network called M-SRN is proposed in this paper. M-SRN can achieve high-quality diverse category reconstruction from a collection of images. The key idea of M-SRN is to use memory representations to enhance reconstruction. Memory representations, which served as high-dimensional feature vectors, were stored in a lightweight memory bank. Its buffering mechanism allows the model to perform contrastive learning across multiple batches, thereby enhancing contextual awareness. In addition, the rich prior knowledge embedded in the memory bank enables features to be clustered into purified representations with distinct semantic meanings. These purified representations are utilized to compensate for intermediate features, thereby strengthening the ability of the model to express complex topological structures.

Specifically, M-SRN consists of foreground perception, reconstruction, and neighbor consistency enhancement modules. An offline segmentation network may produce rough or even failed results in instances with significant domain differences, thereby affecting reconstruction quality. To address this issue, a foreground perception module was designed to optimize offline segmentation results. First, we obtained foreground cues through contrastive learning between foreground and background memory representations. These cues were then used to refine the offline segmentation masks. Owing to the absence of explicit 3D supervision or novel viewpoints, a reconstruction module with feature compensation was developed to enhance dataset-level instance consistency. We purified noisy memory representations to obtain representative memory prototypes for each category. These purified memories were then used to compensate for the encoded features according to their affinity. The aggregated features were decoded to obtain the reconstructed mesh. The feature compensation network enabled the model to identify and utilize common features across different categories, such as the legs of tables and chairs, thereby enhancing its generalization ability. The neighbor consistency enhancement module uses shape-similar images for pseudo-novel view supervision to strengthen the consistency among instances within the same category. To achieve this, a dynamically updated consistency enhancement strategy was proposed. Pseudo-view images are dynamically selected through clusters of memory representations, thereby providing greater flexibility and robustness.

To evaluate M-SRN, we conducted experiments on ShapeNet [22], Pix3D [23] and Pascal3D+ [24]. Extensive experimental results demonstrate that M-SRN outperforms state-of-the-art weakly supervised approaches and achieves competitive performance compared to 2D- and 3D-supervised approaches.

The contributions of this study are as follows.

(1) A novel diverse-category single-view reconstruction framework.

(2) An adaptive context-aware foreground perception module, which effectively refines offline segmentation masks.

(3) A purified memory-based cross-category feature compensation module that enhances dataset-level instance consistency.

(4) A dynamic neighbor consistency enhancement loss, which addresses the ambiguity of single-view reconstruction.

2 Related work

2.1 Single-view 3D reconstruction

Single-view 3D reconstruction methods can be categorized into three groups based on the annotation type: 3D-supervised methods [5, 6, 19, 25–30], 2D-supervised methods [8–13, 17, 18, 31–35], and weakly supervised methods [14–16, 21, 36–40].

3D supervision-based methods. These methods rely on 3D shape annotations. Choy et al. [6] reconstructed 3D voxels using a standard long short-term memory (LSTM) framework. Fan et al. [5] proposed a point-set generation network that marked the first achievement of point-cloud reconstruction. To further enhance the details of the reconstruction, Di et al. [25] used a centered diffusion probabilistic model to achieve point cloud reconstruction. However, these methods are limited to specific categories of reconstruction. Huang et al. [19] proposed a strong regression-based method to achieve zero-shot shape reconstruction.

2D supervision-based methods. These methods aim to replace hard-to-obtain 3D information with 2D annotations. Yan et al. [8] proposed a voxel reconstruction network using multi-view images and the corresponding pose annotations. They employed a differentiable dense sampling layer to project voxels onto a 2D plane. Tulsiani et al. [32] simultaneously learned the poses and shapes of input images, eliminating the need for pose annotations. Navaneet et al. [11] proposed a depth-aware point-feature-rendering module that achieved significant results using

color, part segmentation, and surface normals as annotations. By leveraging advancements in Gaussian point clouds, Szymanowicz et al. [17] proposed a diverse-category Gaussian point-cloud reconstruction network. Similarly, Xu et al. [18] employed a cascaded pipeline to upsample generated Gaussian point clouds to achieve high-quality reconstructions.

Weakly-supervised methods. These methods rely solely on masks and category-specific priors. Navaneet et al. [15] used cycle consistency to reconstruct point clouds. Building on [15], Hu et al. [36] introduced interpolation and landmark consistencies for mesh reconstruction. Monnier et al. [16] proposed a 3D reconstruction network using progressive conditioning and neighbor consistency. They relied on category-specific parameter priors to implement progressive training. To enhance generalization, Huang et al. [21] proposed an SDF reconstruction network based on image-text model consistency, enabling diverse-category reconstruction under weak supervision.

Unlike previous approaches, (1) M-SRN achieves high-quality reconstruction across diverse categories; (2) M-SRN implements dataset-level consistency reconstruction through a lightweight memory bank; (3) M-SRN utilizes an innovative neighbor-selection strategy to enhance the flexibility and robustness of the model-training process.

2.2 Foreground perception

This task aimed to predict the corresponding foreground masks for a sequence of images. Zhou et al. [41] generated class-activation maps (CAM) via a global average pooling layer. The CAM effectively highlighted the key regions that influenced the classification decisions of the model. However, classifiers typically focus on the most distinguishing features of an object for recognition, which results in incomplete outcomes. Xie et al. [42] introduced a two-stage (generation-refinement) framework. They extracted activation maps directly from shallow features using convolutional layers, thereby allowing the activation maps to be refined online. Wu et al. [43] learned a high-quality activation map through the reduction of background pixel activation. Although they achieved effective results, their model relies on the capability of the classification network, limiting its generalization ability on datasets with few categories. Xie et al. [44] solved this problem by performing contrastive learning on image features inside the batch. Their approach did not rely on a classification network, making it more robust. However, their perception capability was limited by the batch size, which not only consumes substantial resources, but also lacks global contrast.

Unlike previous methods, our memory-bank-based foreground perception method offers a stronger contextual awareness and is adaptable to inputs with varying category numbers.

2.3 Neighbor reconstruction

Reconstruction based on neighboring images aims to enhance the shape consistency of instances within the same category during training. The main concept is to find a pseudo-view supervision image with a shape similar to that of the input image. Navaneet et al. [15] utilized a classification network pre-trained on ImageNet to extract features from input image sequences. Images with latent codes that closely matched the input were identified as neighboring images. Monnier et al. [16] recorded the input and intermediate codes via a memory bank during training. Shape-similar images are identified by comparing the codes of the input images with those stored in the memory bank. Although they achieved a dynamic selection of pseudo-view images, their accuracy depended heavily on the quality of the encoder used in the training. With the advancement of image-text models, Huang et al. [21] leveraged CLIP [45] to obtain latent codes for images, resulting in a significant improvement in accuracy. However, Refs. [15, 21] required fixing similar images before training. Incorrectly predefined images can negatively affect model training, leading to suboptimal reconstruction results.

Unlike previous methods, we leveraged memory prototypes to allow the model to autonomously select suitable pseudo-view images from CLIP-predicted candidates during training, making the process more flexible and accurate.

3 Method

The overall structure of M-SRN is shown in Figure 1. In the preprocessing stage, a pretrained segmentation network [46] was first used to obtain coarse masks. Foreground cues were then obtained using the foreground perception module (detailed in Subsection 3.1) to refine the coarse masks. We follow [21] to assign a corresponding neighbor set $\mathcal{I}_{\text{can}}$ to each input image, which is then used for subsequent neighbor reconstruction. In the training stage, the segmented image I^r is encoded into the shape code z_s, texture code z_t, and pose code z_c, which are defined as follows:

$$z_s = E^s_{\text{re}}(I^r, \theta^s_e), \tag{1}$$

$$z_t = E^t_{\text{re}}(I^r, \theta^t_e), \tag{2}$$

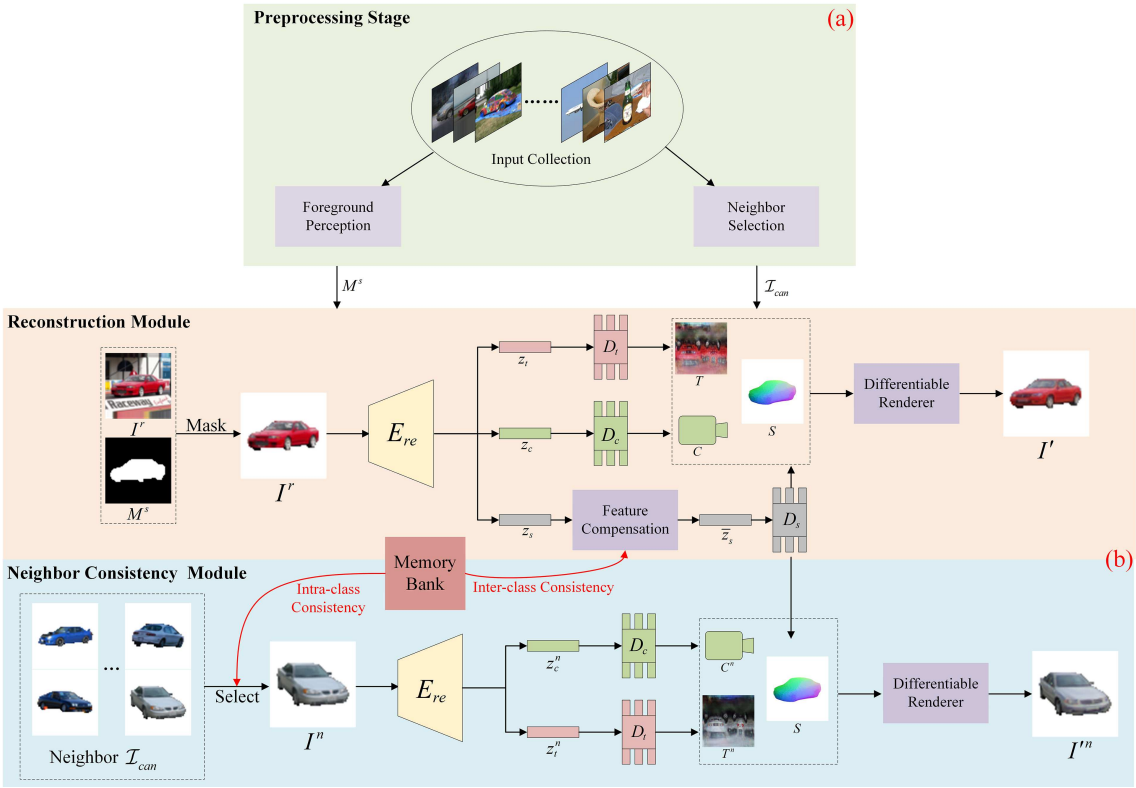

Figure 1 (Color online) M-SRN architecture. (a) Preprocessing stage. Masks corresponding to the image collections are first predicted via the proposed foreground perception module. Then, the candidate neighbor image set $\mathcal{I}_{\text{can}}$ is selected. (b) Training stage. A segmented input image I^r is encoded and linearly transformed to obtain latent vectors $[z_s, z_t, z_c]$. To enhance shape consistency among instances of different categories, z_s is input into the feature compensation module to produce $\overline{z}_s$. These vectors are then fed into decoders $[D_s, D_t, D_c]$ to generate shape S, texture T, and camera pose C. The differentiable renderer π is used to convert S, T, and C into a 2D projection I'. Similarly, the dynamically selected neighbor image I^n is fed into a shared encoder-decoder network to obtain the projection I'^n. The discrepancy between I' and I^r is used as self-supervised reconstruction loss, while that between I^n and I'^n is used as the neighbor enhancement loss.

$$z_c = E_{\text{re}}^c(I^r, \theta_e^c), \tag{3}$$

where E_{re}^s, E_{re}^t and E_{re}^c denote the shape, texture, and pose encoders, respectively. We utilize category memory knowledge to compensate for the shape encoding z_s (as detailed in Subsection 3.2), resulting in an enhanced feature vector $\bar{z}_s$. The reconstructed texture T and pose C are obtained using the corresponding decoders, which are defined as

$$T = D_t(z_t, \theta_d^t), \tag{4}$$

$$C = D_c(z_c, \theta_d^c), \tag{5}$$

where D_t is the texture decoder composed of convolutional layers and D_c is the pose decoder composed of linear layers. The enhanced feature is fed into a multilayer perceptron D_s to predict ellipse deformation. The reconstructed mesh S is defined as follows:

$$S = X + D_s(\bar{z}_s, X, \theta_d^s), \tag{6}$$

where X denotes the 3D vertex of the ellipsoid. Because calculating the loss only from the input viewpoint degrades the shape, the neighbor module is designed to alleviate the ambiguity of single-view supervision. The high-dimensional feature clusters of each category in the memory bank were used to dynamically select a set of pseudo-viewpoint supervision images from $\mathcal{I}_{\text{can}}$ (detailed in Subsection 3.3). We used the discrepancy between I^n and I'^n as an additional constraint to optimize the overall network (detailed in Subsection 3.4).

3.1 Foreground perception module

The foreground perception module extracts foreground information from input images. Previous methods [14, 19] have relied on offline models to produce coarse masks. In the following text, we use 'off-the-shelf masks' to refer to the masks obtained from a pretrained segmentation network. However, for images with complex backgrounds, off-the-shelf masks often exhibit significant noise artifacts, leading to erroneous loss calculations. To address this

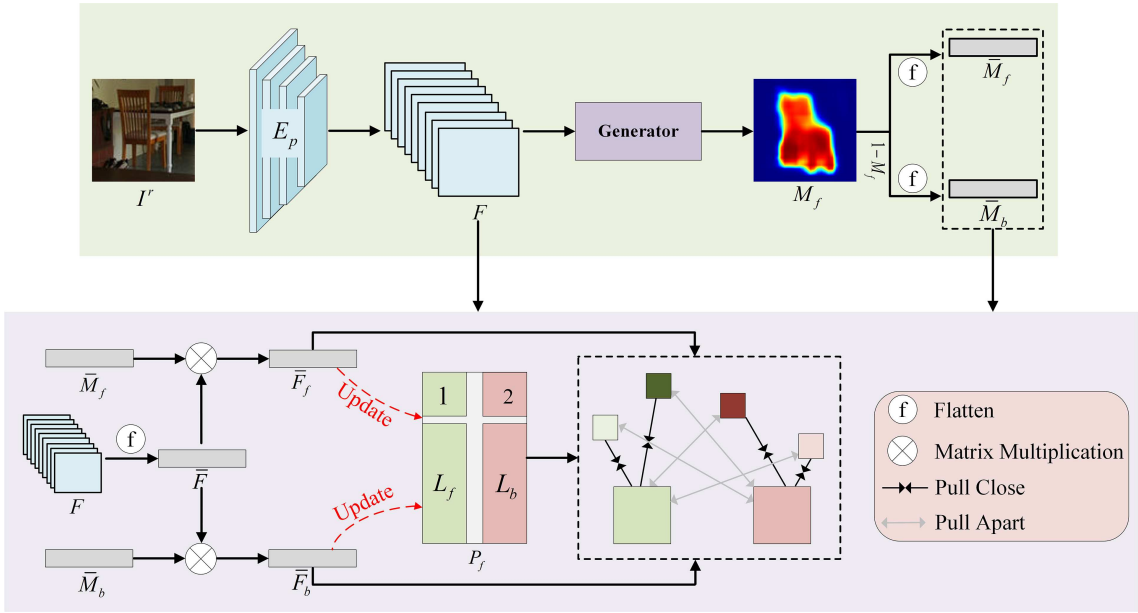

Figure 2 (Color online) Memory bank-based contrastive learning module.

issue, we designed a foreground cue generation method based on a memory bank. These cues were used to filter out noise from off-the-shelf masks.

Our foreground cue generation module is based on a previous study [43]. We aimed to fine-tune the ILSVRC [47] pretrained model for our input data. To achieve this, an additional contrastive loss based on feature representation was employed during the fine-tuning process, as illustrated in Figure 2. Specifically, for input image I^r, we extract its foreground cues M_f. To do this, I^r is first fed into the encoder E_p to obtain the feature map F. Then, a generator with a convolution layer Conv and a sigmoid layer Sig takes F as the input to generate foreground cues M_f and background cues M_b, which can be formulated as

$$M_f = \mathrm{Sig}(\mathrm{Conv}(F, \theta_f)), \tag{7}$$

$$M_b = 1 - M_f, \tag{8}$$

where θ_f represents the parameters of the convolutional layers. To facilitate the subsequent contrastive learning, $[F, M_f, M_b]$ are flattened into $\bar{F} \in \mathbb{R}^{Q_f \times HW}$, $\bar{M}_f \in \mathbb{R}^{1\times HW}$ and $\bar{M}_b \in \mathbb{R}^{1\times HW}$, respectively. The foreground features $\bar{F}_f$ and background features $\bar{F}_b$ are then computed as follows:

$$\bar{F}_f = \bar{M}_f \otimes \bar{F}^{\mathrm{Tr}}, \tag{9}$$

$$\bar{F}_b = \bar{M}_b \otimes \bar{F}^{\mathrm{Tr}}, \tag{10}$$

where $\otimes$ and Tr represent matrix multiplication and transposition, respectively. $\bar{F}_f$ and $\bar{F}_b$ are used for contrastive learning with the memory representations in the bank. The foreground perceptual memory bank P_f is composed of a foreground term L_f and background term L_b, which is defined as

$$P_f = \{L_f, L_b\}, \tag{11}$$

$$L_f, L_b \in \mathbb{R}^{K\times Q_f}, \tag{12}$$

where K is the number of feature vectors in each term and Q_f is the dimensionality of each feature. L_f and L_b store the foreground and background features recorded during the training, respectively. For $\bar{F}_f$, we strengthened its alignment with set $\{f^+ \in L_f\}$, While also separating it from set $\{f^- \in L_b\}$. For $\bar{F}_b$, we improved its similarity with the vectors set $\{f_b^+ \in L_b\}$, while distancing it from the key vectors set $\{f_b^- \in L_f\}$.

This process is implemented via loss $\mathcal{L}_{\mathrm{for}}$, which is represented as

$$\mathcal{L}_{\mathrm{for}}^f = \frac{1}{K}\sum_{f^+\in L_f} -\log \frac{\mathrm{e}^{\mathrm{sim}(\bar{F}_f, f^+)/\gamma}}{\mathrm{e}^{\mathrm{sim}(\bar{F}_f, f^+)/\gamma} + \sum_{f^-\in L_b} \mathrm{e}^{\mathrm{sim}(\bar{F}_f, f^-)/\gamma}}, \tag{13}$$

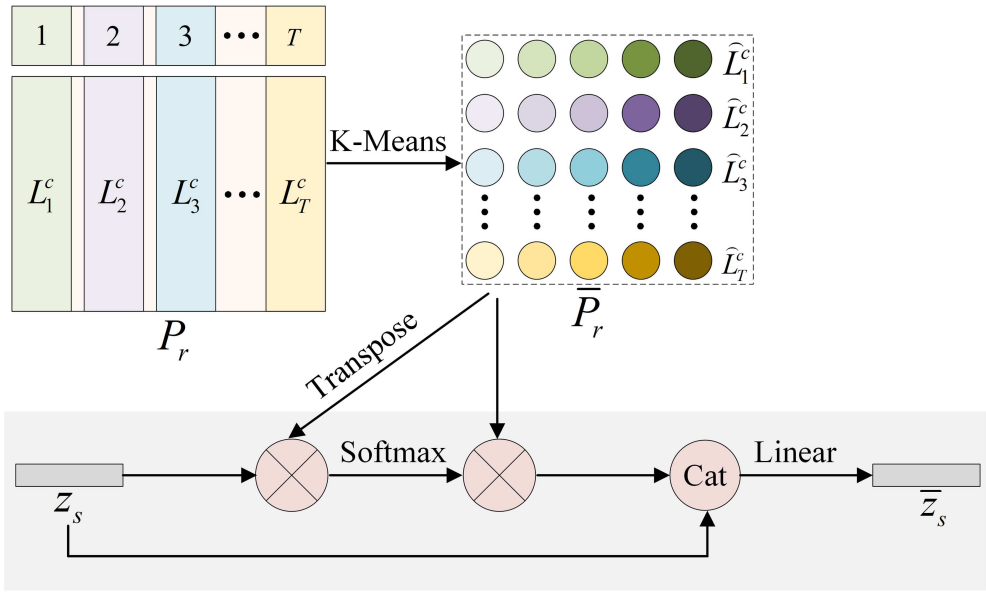

Figure 3 (Color online) Feature compensation module.

$$\mathcal{L}_{\text{for}}^{b}=\frac{1}{K}\sum_{f_b^+\in L_b}-\log\frac{\mathrm{e}^{\text{sim}(\bar{F}_b,f_b^+)/\gamma}}{\mathrm{e}^{\text{sim}(\bar{F}_b,f_b^+)/\gamma}+\sum_{f_b^-\in L_f}\mathrm{e}^{\text{sim}(\bar{F}_b,f_b^-)/\gamma}}, \tag{14}$$

$$\mathcal{L}_{\text{for}}=\mathcal{L}_{\text{for}}^{f}+\mathcal{L}_{\text{for}}^{b}, \tag{15}$$

where γ is a temperature hyperparameter and sim denotes the cosine similarity.

We utilized the foreground features $\bar{F}_f$ and background features $\bar{F}_b$ extracted from the input image to update the L_f and L_b items in the memory bank. A smoother update approach was used for P_r, which is defined as

$$l_v^f\leftarrow\eta l_v^f+(1-\eta)\bar{F}_f, \tag{16}$$

$$l_v^b\leftarrow\eta l_v^b+(1-\eta)\bar{F}_b, \tag{17}$$

where η is the momentum of the memory evolution. l^f and l^b are feature vectors stored in L_f and L_b, respectively. v is a pointer value ranging from one to K, which marks the position of the feature vector to be updated in the memory bank. To ensure its effectiveness, the memory representation is updated only when the category prediction scores (detailed in [43]) exceed δ_f.

After obtaining the foreground cue M_f, we removed noise from the off-the-shelf masks M_o to produce a refined mask M_s. U2-Net [46] was used as an offline segmentation network to obtain a rough mask M_o. Pixels in M_f with activation values below the threshold λ were identified as noise and were subsequently removed from the mask M_o. The final refined mask M^s and corresponding segmented image are defined as follows:

$$M_s(i,j)=\begin{cases}M_o(i,j), & \text{if } M_f(i,j)\geqslant\lambda,\\ 0, & \text{if } M_f(i,j)<\lambda,\end{cases} \tag{18}$$

$$I^r=I^r\times M^s, \tag{19}$$

where i and j represent the pixel coordinates.

3.2 Feature compensation module

In weakly supervised diverse-category single-view reconstruction, the absence of explicit shape supervision often results in significant detail loss when handling complex topologies. In addition, the model fails to effectively learn the features of categories with sparse samples, resulting in poor reconstruction quality. To address this issue, a feature compensation module based on purified memory is proposed, as shown in Figure 3. By exploring the consistency between different instances (e.g., the legs of tables and chairs and the tyres of motorcycles and bicycles), M-SRN achieves detail-aware SVR.

Specifically, we first established a memory representation bank P_r to store the shape encodings of the input instances. Unlike the memory bank P_f used for foreground perception, memory bank P_r stores memory representations for each category and is defined as

$$P_r = \{L_1^c, L_2^c, \ldots, L_T^c\}, \tag{20}$$

$$L_i^c \in \mathbb{R}^{K \times Q_r},\ i \in \{1, 2, \ldots, T\}, \tag{21}$$

where Q_r is the dimensionality of the features and T represents the number of categories. We used memory representations in the bank to compensate for the latent code z_s of the input image. However, the memory vectors stored in the memory bank P_r contain considerable noise and irrelevant information. To address this issue, L^c was purified into K' representative prototype representations using k-means clustering at the start of each epoch. The purified memory bank P_r' is defined as

$$P_r' = \left\{\widehat{L_1^c}, \widehat{L_2^c}, \ldots, \widehat{L_T^c}\right\}, \tag{22}$$

$$\widehat{L_i^c} \in \mathbb{R}^{K' \times Q_r},\ i \in \{1, 2, \ldots, T\}. \tag{23}$$

P_r' effectively filters out outliers and redundant features while retaining the most representative feature prototypes in each category. To leverage the information in the memory bank related to image features for compensation, we must calculate the affinity A between the latent encoding z_s of the input image and the purified representations P_r' of different categories. A was used to perform a weighted fusion of the prototype features, enabling correlation-driven feature integration. Using this similarity score A, the model can select the purified memories that are most relevant to the input to compensate for intermediate features. To achieve this, P_r' is transformed into a tensor $\bar{P}_r$ in the shape $T \times K' \times Q_r$. A is defined as

$$A = \mathrm{sof}(z_s \otimes \bar{P}_r^{\mathrm{Tr}}), \tag{24}$$

where $\otimes$ represents the matrix multiplication. sof represents the softmax operation that normalizes each input row. $\bar{P}_r^{\mathrm{Tr}}$ indicates the transpose of the matrix $\bar{P}_r$. Based on the affinity score A, the compensation feature z_{s+} is calculated as

$$z_{s+} = A \otimes \bar{P}_r. \tag{25}$$

The final shape feature $\overline{z}_s$ is defined as follows:

$$\bar{z}_s = \mathrm{Lin}([z_s, z_{s+}]), \tag{26}$$

where Lin is the linear layer used to reduce the dimensions back to Q_r.

After obtaining the compensated features, we updated P_r using the same method as P_f. To reduce the noise in the memory bank, a confidence score S^r was obtained by calculating the mask IoU between I' and I^r. The memory representation is updated only if the score S^r exceeds δ_r.

3.3 Neighbor consistency enhancement module

Training solely with the loss between the projection map and the input image from the input perspective leads to a degenerated result. This is because supervision from a single perspective only ensures that the reconstruction is correct from that perspective, without guaranteeing accuracy from other perspectives. The neighbor consistency enhancement module aims to improve the consistency among instances within the same category by using shape-similar neighbor images. It introduces a neighborhood loss calculated using similar images on top of the original input loss. Because the input image and similar images share the same shape, the neighborhood loss computed from the shape reconstructed using the input image can effectively establish multi-view constraints. Previous methods [15, 21] utilized pretrained classification networks to obtain latent encodings of images. By comparing the similarity of latent codes, they assigned neighboring images to each input image before training. However, this fixed assignment method lacks robustness to incorrect neighbor images, which affects the reconstruction accuracy. To address this issue, we proposed a dynamic neighbor-consistency reconstruction method based on purified memory.

Given an input image I^r, we identified candidate similar image sets $\mathcal{I}_{\mathrm{can}}$ during the preprocessing stage. $\mathcal{I}_{\mathrm{all}}$ is the set of input images to be reconstructed, excluding I^r, which is defined as follows:

$$\mathcal{I}_{\mathrm{all}} = \{I_1, I_2, \ldots, I_{\mathrm{Num}}\}, \tag{27}$$

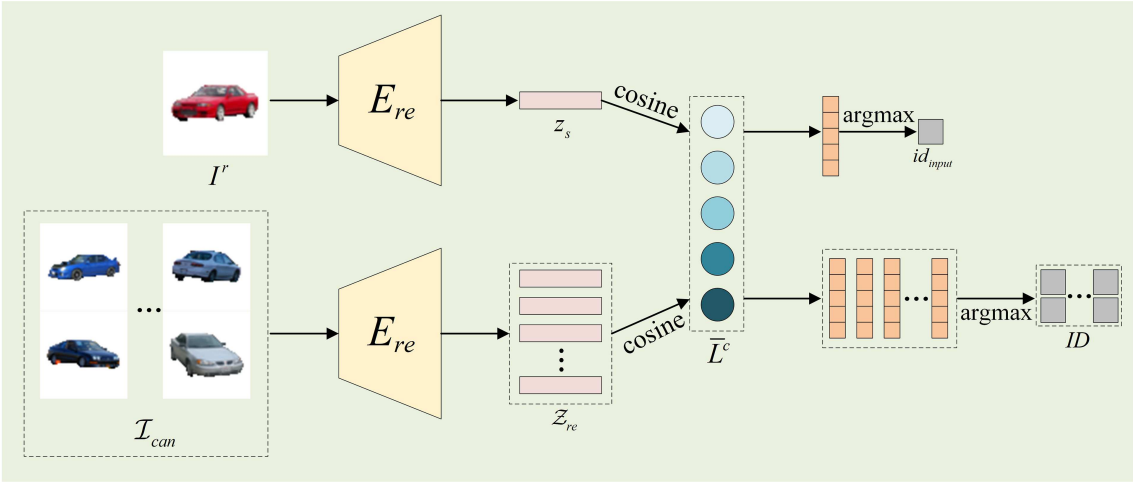

Figure 4 (Color online) Dynamic neighbor selection module.

where Num denotes the number of images in $\mathcal{I}_{\text{all}}$. To search for $\mathcal{I}_{\text{can}}$ in the input set $\mathcal{I}_{\text{all}}$, we leveraged a pretrained CLIP [45] network N^{clip} to extract latent code for both I^r and images in $\mathcal{I}_{\text{all}}$. The latent code $z_{\text{inp}}^{\text{clip}}$ corresponding to image I^r and the latent code set $\mathcal{Z}_{\text{clip}}$ corresponding to set $\mathcal{I}_{\text{all}}$ are defined as follows:

$$z_{\text{inp}}^{\text{clip}} = N^{\text{clip}}(I^r), \tag{28}$$

$$z_j^{\text{clip}} = N^{\text{clip}}(I_j), \tag{29}$$

$$\mathcal{Z}_{\text{clip}} = \{z_1^{\text{clip}}, \ldots, z_j^{\text{clip}}, \ldots, z_{\text{Num}}^{\text{clip}}\}. \tag{30}$$

To measure the similarity between the input image I^r and each image I_j in the collection $\mathcal{I}_{\text{all}}$, we calculated the cosine similarity between their latent codes:

$$\cos(z_{\text{inp}}^{\text{clip}}, z_j^{\text{clip}}) = \frac{z_{\text{inp}}^{\text{clip}} \cdot z_j^{\text{clip}}}{\| z_{\text{inp}}^{\text{clip}} \| \cdot \| z_j^{\text{clip}} \|}. \tag{31}$$

We then selected the top $K_{\cos}$ images from $\mathcal{I}_{\text{all}}$ images with the highest cosine similarity scores, thus forming the candidate set $\mathcal{I}_{\text{can}}$:

$$\mathcal{I}_{\text{can}} = \left\{I_1^n, I_2^n, \ldots, I_{k_{\cos}}^n\right\}. \tag{32}$$

After obtaining the preselected shape-similar image set $\mathcal{I}_{\text{can}}$, we allowed the model to filter the images autonomously during training for consistent reconstruction. Figure 4 shows the details of the dynamic selection module. We utilized high-dimensional memory clusters to assess the similarity between the candidate image set and the input image, thereby dynamically filtering neighboring images. In this process, we shared prior knowledge stored in the memory bank of the feature compensation module. Specifically, the purified memory bank P_r' is further refined to obtain P_r''. Compared to P_r', P_r'' contains K'' more representative prototype memory representations for each category, which is defined as

$$P_r'' = \{\bar{L}_1^c, \bar{L}_2^c, \ldots, \bar{L}_T^c\}, \tag{33}$$

$$\bar{L}_i^c \in \mathbb{R}^{K'' \times Q_r},\ i \in \{1, 2, \ldots, T\}. \tag{34}$$

The candidate image set $\mathcal{I}_{\text{can}}$ is processed through the encoder of the reconstruction module to obtain their corresponding latent code set $\mathcal{Z}_{\text{re}} = \left\{z_1^{\text{re}}, z_2^{\text{re}}, \ldots, z_{k_{\cos}}^{\text{re}}\right\}$. By calculating the cosine similarity between z_s and the K'' prototypes of the corresponding category, we obtain the prototype domain indices id_{input}:

$$\text{id}_{\text{input}} = \arg\max_j \cos(z_s, \bar{l}_j^c), \tag{35}$$

where $\bar{l}_j^c$ denotes the jth prototype of the corresponding category item $\bar{L}^c$. Similarly, the set of prototype domain indices ID corresponding to the images in $\mathcal{I}_{\text{can}}$ is defined as follows:

$$\text{id}_i = \arg\max_j \cos(z_i^{\text{re}}, \bar{l}_j^c), \tag{36}$$

$$\mathrm{ID} = \{\mathrm{id}_1, \mathrm{id}_2, \ldots, \mathrm{id}_{k_{\cos}}\}. \tag{37}$$

The filtered candidate image set $\mathcal{I}_{\mathrm{re}}$ is defined as

$$\mathcal{I}_{\mathrm{re}} = \{I_x^n | \mathrm{id}_x = \mathrm{id}_{\mathrm{input}}\},\ x \in \{1, 2, \ldots, N_{\mathrm{re}}\}, \tag{38}$$

where N_{re} is the number of filtered neighboring images. Through dynamic filtering during training, our approach retains advantage of CLIP in neighbor selection while enhancing the fault tolerance of the model, leading to improved reconstruction accuracy.

Because the neighbor image I^n shares similar shapes with the input image I^r, shape S reconstructed from I^r should also be applicable to I^n. Therefore, we utilize the shape constraint between the reconstructed shape S and neighboring image I^n to provide a novel perspective of supervision for the reconstruction module. Specifically, the neighboring image I^n in $\mathcal{I}_{\mathrm{re}}$ is input into the reconstruction module to produce the texture T^n and pose C^n. S, T^n and C^n are fed into the differentiable projection π to produce a neighboring projection I'^n. In our experiments, SoftRasterizer [10] was selected as the rendering model π. The discrepancy between I^n and I'^n is utilized to enhance the consistency of the neighbor. Note that we do not use texture consistency reconstruction in the neighbor reconstruction. This is because the introduction of texture consistency does not improve the shape prediction accuracy and instead increases the training cost. We believe that this is because of our explicit mask supervision and flexible neighbor-selection module, which effectively enhances the accuracy of shape prediction, making the inclusion of texture consistency unnecessary.

3.4 Loss function

$\mathcal{L}_{\mathrm{for}}$ is used to optimize the foreground perception module. During the training of the reconstruction network, our loss function $\mathcal{L}_r$ consists of the reconstruction loss $\mathcal{L}_{\mathrm{rec}}$ and neighbor reconstruction loss $\mathcal{L}_{\mathrm{ne}}$. $\mathcal{L}_{\mathrm{rec}}$ calculates the difference between the input image I^r and projection I', which is defined as

$$\mathcal{L}_{\mathrm{rgb}}(I^r, I') = \frac{1}{hw} \sum_{i,j} \parallel I_{i,j}^r - I_{i,j}' \parallel, \tag{39}$$

$$\mathcal{L}_{\mathrm{bce}}(M^s, M') = \frac{1}{hw} \sum_{i,j} -M_{i,j}^s \log M_{i,j}' - (1 - M_{i,j}^s) \log(1 - M_{i,j}'), \tag{40}$$

$$\mathcal{L}_{\mathrm{aff}}(M^s, M') = \sum_{i,j} \min_{(k,l) \in M_+^s} ((i-k)^2 + (j-l)^2) M_{i,j}' M_{k,l}^s + \sum_{i,j} \min_{(k,l) \in M_+'} ((i-k)^2 + (j-l)^2) M_{i,j}^s M_{k,l}', \tag{41}$$

$$\mathcal{L}_{\mathrm{rec}} = \mathcal{L}_{\mathrm{bce}}(M^s, M') + \varepsilon \times \mathcal{L}_{\mathrm{aff}}(M^s, M') + \mathcal{L}_{\mathrm{rgb}}(I^r, I'), \tag{42}$$

where M^s and M' are foreground masks corresponding to I^r and I'. h and w correspond to the length and width of the input image, respectively. M_+^s and M_+' represent the sets of pixel indices in M^s and M' where the values are nonzero. $\mathcal{L}_{\mathrm{bce}}$ measures the discrepancy between the predicted and ground-truth masks using log-based regularization terms. $\mathcal{L}_{\mathrm{aff}}$ was designed to measure the difference between the predicted and input masks. It is computed by evaluating the spatial and value differences between the predicted and actual masks, ensuring that the model predicts not only the correct mask location but also the correct mask values. $\mathcal{L}_{\mathrm{ne}}$ calculates the difference between neighboring images in $\mathcal{I}_{\mathrm{re}}$ and their corresponding projected images, which is defined as

$$\mathcal{L}_x^n = \mathcal{L}_{\mathrm{bce}}(M_x^n, M_x'^n) + \varepsilon \times \mathcal{L}_{\mathrm{aff}}(M_x^n, M_x'^n) + \mathcal{L}_{\mathrm{rgb}}(I_x^n, I_x'^n), \tag{43}$$

$$\mathcal{L}_{\mathrm{ne}} = \frac{1}{N_{\mathrm{re}}} \sum_{x=1}^{N_{\mathrm{re}}} \mathcal{L}_x^n, \tag{44}$$

where x represents the index of the neighboring images. $\mathcal{L}_{\mathrm{bce}}$ and $\mathcal{L}_{\mathrm{aff}}$ compute the mask loss between the projection map and neighboring image, whereas $\mathcal{L}_{\mathrm{rgb}}$ calculates the RGB loss between them. The overall loss $\mathcal{L}_r$ of the reconstruction module is defined as follows:

$$\mathcal{L}_r = \mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{ne}} + \mathcal{L}_{\mathrm{normal}} + \mathcal{L}_{\mathrm{lap}}, \tag{45}$$

where $\mathcal{L}_{\mathrm{normal}}$ [48] ensures mesh smoothness through aligning the neighboring faces, and $\mathcal{L}_{\mathrm{lap}}$ [49] averages the positions of vertices with those of their neighbors to reduce mesh noise.

Table 1 Network parameter values.

K	K'	K''	λ	$k_{\cos}$	γ	η	ε	δ_f	δ_r
500	10	5	0.4	6	0.99	0.99	1e−6	0.7	0.5

4 Experiments

4.1 Datasets

We evaluated M-SRN using the following datasets: ShapeNet [22], Pix3D [23] and Pascal3D+ [24].

ShapeNet. ShapeNet is a large-scale 3D object dataset covering fifty-five object categories with extensive shapes and semantic annotations. We followed [50] to generate rendered images for 13 categories as a dataset.

Pix3D. Pix3D is a dataset that pairs real-world images with accurate 3D models and is designed to advance research on 3D reconstruction and image-to-model alignment. We used the train/val/test splits provided in [21].

Pascal3D+. Pascal3D+ is a dataset that provides images with 3D pose annotations across twelve rigid object categories. It features a more complex topology and image background. We followed [31] to generate the train/test splits.

4.2 Evaluation metrics

To comprehensively evaluate the performance of M-SRN, we utilized three common metrics: chamfer distance (CD) [5], F-score [21, 25], and 3D IoU [31]. The advantage of CD lies in its ability to accurately quantify global geometric differences between the reconstructed shape and ground truth. The chamfer distance between the ground truth and predicted shapes is defined as

$$d_{\mathrm{CD}}\left(S_1, S_2\right) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{x \in S_2} \min_{y \in S_1} \|x - y\|_2^2 . \tag{46}$$

In contrast, the F-score combines precision and recall, effectively assessing the accuracy of local matches and providing a holistic measure of the quality of point-cloud reconstruction. The F-score was defined as follows:

$$\text{F-score} = \frac{2 \times \mathrm{Pre} \times \mathrm{Re}}{\mathrm{Pre} + \mathrm{Re}}, \tag{47}$$

where Pre is the proportion of correctly predicted positive instances out of the total predicted positive instances and Re is the proportion of correctly predicted positive instances out of the total actual positive instances. 3D IoU was compared with the baseline methods for single-category SVR. In our experiments, ICP [51] is used to align the sampled point clouds.

4.3 Implementation details

Experiments were performed at a 64×64 image resolution. We followed [16] by using a sphere as the reconstruction target to pre-train the reconstruction module. For real-world datasets, we used well-trained network parameters from ShapeNet for initialization. To facilitate the effective learning of poses, we fixed the size of the latent codes z_s and z_t to 1 and 2 before 10k iterations. The neighbor reconstruction module was applied after 10k iterations. The memory bank was initialized using a random tensor from a standard normal distribution. The feature compensation module was not applied during training of the first epoch. The M-SRN model was developed in PyTorch and trained on a GeForce RTX 3090 GPU, with a batch size of 16. The training process used the Adam optimizer, beginning with a learning rate of 1e−4. The parameter settings are listed in Table 1.

4.4 Comparison of results

In this subsection, M-SRN is compared with leading SVR methods on both synthetic and real-world datasets. We use the following notation to represent the type of supervision required by each method: 'M' for multi-view, 'K' for keypoints, 'C' for pose, 'S' for silhouette, 'A' for assumption, and 'P' for prior. 'I' is additionally used to denote the method that uses only the input RGB image as the supervision condition.

Table 2 Diverse-category quantitative comparison results on ShapeNet. A downward arrow (↓) indicates that a lower value is desirable, whereas an upward arrow (↑) signifies that a higher value is preferable.

	Supervision	Chamfer distance↓	F-score@0.01↑
ShapeClipper [21]	S+A	3.72	0.169
Unicorn [16]	I	2.83	0.216
Splatter [17]	M+C+S	1.32	0.434
Ours	S	1.61	0.365

Table 3 Diverse-category quantitative comparison results on Pix3D.

	Supervision	Chamfer distance↓	F-score@0.01↑
SSMP [40]	S	6.63	0.105
ShapeClipper [21]	S+A	5.58	0.129
One2345 [35]	M+C+S	3.89	0.149
OpenLRM [34]	M+C+S	4.02	0.153
ZeroShape [19]	3D	2.86	0.208
Ours	S	3.29	0.185

4.4.1 *Diverse-category reconstruction on ShapeNet*

We compared our approach with the state-of-the-art 2D-supervised and weakly supervised methods using the ShapeNet dataset. The code provided was used to train the diverse-category baseline models. Because Splatter reconstructs sparse Gaussian point clouds, we sampled the point clouds based on opacity. Because the synthetic datasets had inherently white backgrounds, our foreground perception module was not utilized in ShapeNet. The results of the quantitative comparisons are presented in Table 2 [16, 17, 21]. Overall, M-SRN outperformed the weakly supervised methods and achieved a performance similar to that of Splatter. Although mesh representations have limitations in modeling objects with complex structures, our method remains competitive with Gaussian point clouds and SDF-based representations when applied to synthetic datasets with relatively regular topological structures. The qualitative results are shown in Figure 5. In the baseline methods, Unicorn [16] confuses categories and produces degraded results. Unicorn relies on the quality of the encoder to select pseudo-view supervision during training and lacks precise and clear neighbor constraints. In contrast, our neighbor constraint method eliminates dependence on the encoder, thereby preventing shape degradation. ShapeClipper [21] predicts consistent shapes but performs poorly on details such as the legs of the chairs. Compared with ShapeClipper, our feature compensation module enables the selection of the most relevant global features to enhance detail representation, thereby achieving superior capture of both global shapes and finer details.

4.4.2 *Diverse-category reconstruction on Pix3D*

To verify the generalization ability of real-world data, M-SRN was evaluated on the Pix3D dataset, which featured diverse shapes, textures, and environments. Qualitative and quantitative comparisons are presented in Figure 6 and Table 3 [19,21,34,35,40]. Despite the increased complexity of real-world topological shapes, M-SRN still outperforms weakly supervised approaches and achieves competitive performance compared to 3D-supervised approaches. Owing to the introduction of the global consistency module, M-SRN maintains high-precision predictions for several low-sample categories within the Pix3D dataset. We observed that ShapeClipper predicted uneven surfaces and tended to lose some of the structural information. Compared with ShapeClipper, our mesh-based encoder-decoder architecture enables smoother reconstructions. In addition, our flexible selection of pseudo-view images within high-dimensional clusters reduces the detrimental effects of ShapeClipper's predefined errors, thereby enhancing the overall detail of the reconstruction. However, owing to the inherent initial connectivity of the 3D mesh representation, M-SRN struggles to accurately capture hollow spaces in categories such as chairs and bookcases. Consequently, our performance lags behind that of ZeroShape.

4.4.3 *Diverse-category reconstruction on Pascal3D+*

We conducted experiments on the Pascal3D+ dataset, which included more categories and complex backgrounds. The quantitative comparison results are presented in Table 4 [19, 21, 30, 34, 35]. Our method achieved better performance on more challenging in-the-wild images. The qualitative results are shown in Figure 7. ShapeClipper often predicts more generalized or average shapes when reconstructing complex structures, leading to a significant loss of fine details. Although OpenLRM predicts relatively accurate shapes, it suffers from deformation and degradation in

Figure 5 (Color online) Diverse-category visual comparison results on ShapeNet.

Table 4 Diverse-category quantitative comparison results on Pascal3D+.

	Supervision	Chamfer distance↓	F-score@0.01↑
ShapeClipper [21]	S+A	5.83	0.120
One2345 [35]	M+C+S	4.31	0.146
OpenLRM [34]	M+C+S	4.18	0.150
Transfer [30]	3D	2.29	–
ZeroShape [19]	3D	3.94	0.157
Ours	S	3.41	0.173

certain instances. As a zero-shot model, ZeroShape is trained using a large amount of synthetic data. This makes it overly reliant on the visible parts of an image for understanding novel real-world scenarios. In addition, it tends to generate degraded shapes in certain instances, which can be attributed to erroneous predictions of intermediate representations. Our weakly supervised approach allows for the optimization of unlabeled datasets, thereby enhancing domain adaptability. Furthermore, it improves the model's ability to infer unseen regions, thereby maintaining the integrity of the predicted shapes.

4.4.4 *Single-category reconstruction*

To further demonstrate its effectiveness, M-SRN was compared with the leading single-category SVR methods on ShapeNet and Pascal3D+ Car. A single model was used to train and evaluate a single category. The quantitative comparison results for ShapeNet are listed in Table 5 [13, 16, 17, 21, 52]. For the weakly supervised methods ShapeClipper and Unicorn, we achieved better performance across most categories. In the categories of cars and chairs, M-SRN outperformed existing weakly supervised approaches and achieved results comparable to those of the 2D-supervised method, Splatter. A comparison of weakly-supervised methods on the Pascal3D+ Car dataset

Figure 6 (Color online) Diverse-category visual comparison results on Pix3D.

Table 5 Single-category quantitative comparison results on ShapeNet.

Method	Supervision	Chamfer distance ↓												
		airplane	bench	cabinet	car	chair	display	lamp	phone	rifle	sofa	speaker	table	vessel
TARS [13]	S+C	1.25	–	–	1.48	2.55	–	–	–	–	–	–	–	–
DVR [52]	M+C+S	1.11	1.76	1.58	1.53	2.05	1.63	2.81	0.76	0.83	1.60	2.15	2.30	1.51
Unicorn [16]	I	1.10	1.59	1.37	1.68	2.53	2.20	5.23	1.27	0.97	1.92	2.24	2.43	1.55
ShapeClipper [21]	S+A	1.81	2.15	1.25	2.14	4.56	3.07	5.84	1.58	1.20	3.13	2.94	3.96	1.73
Splatter [17]	M+C+S	–	–	–	1.15	1.48	–	–	–	–	–	–	–	–
Ours	S	0.91	1.21	1.52	1.41	1.89	1.72	3.81	1.47	1.13	1.62	1.96	2.03	1.42

is presented in Table 6 [14, 16, 21, 31, 37, 38]. M-SRN surpasses existing weakly supervised reconstruction methods and achieves high-quality single-category SVR under in-the-wild image conditions. These results demonstrate that our dynamic neighbor module provides effective pseudo-view supervision for input images, even when predicting a single category. Our consistency module enables our method to achieve notable single-category inference results for both synthetic and real-world datasets.

4.5 Ablation experiments

4.5.1 *Effectiveness analysis*

Foreground perception module. We validated the effectiveness of the foreground perception module on Pascal3D+. The qualitative results are shown in Figure 8. Owing to the differences in image feature distributions across domains, the offline segmentation method [46] predicts erroneous masks with noise in some instances. The activation map obtained in [43] highlights rough foreground regions, leading to suboptimal denoising results. Using the activation maps generated by our method to denoise off-the-shelf masks, we achieved results that closely

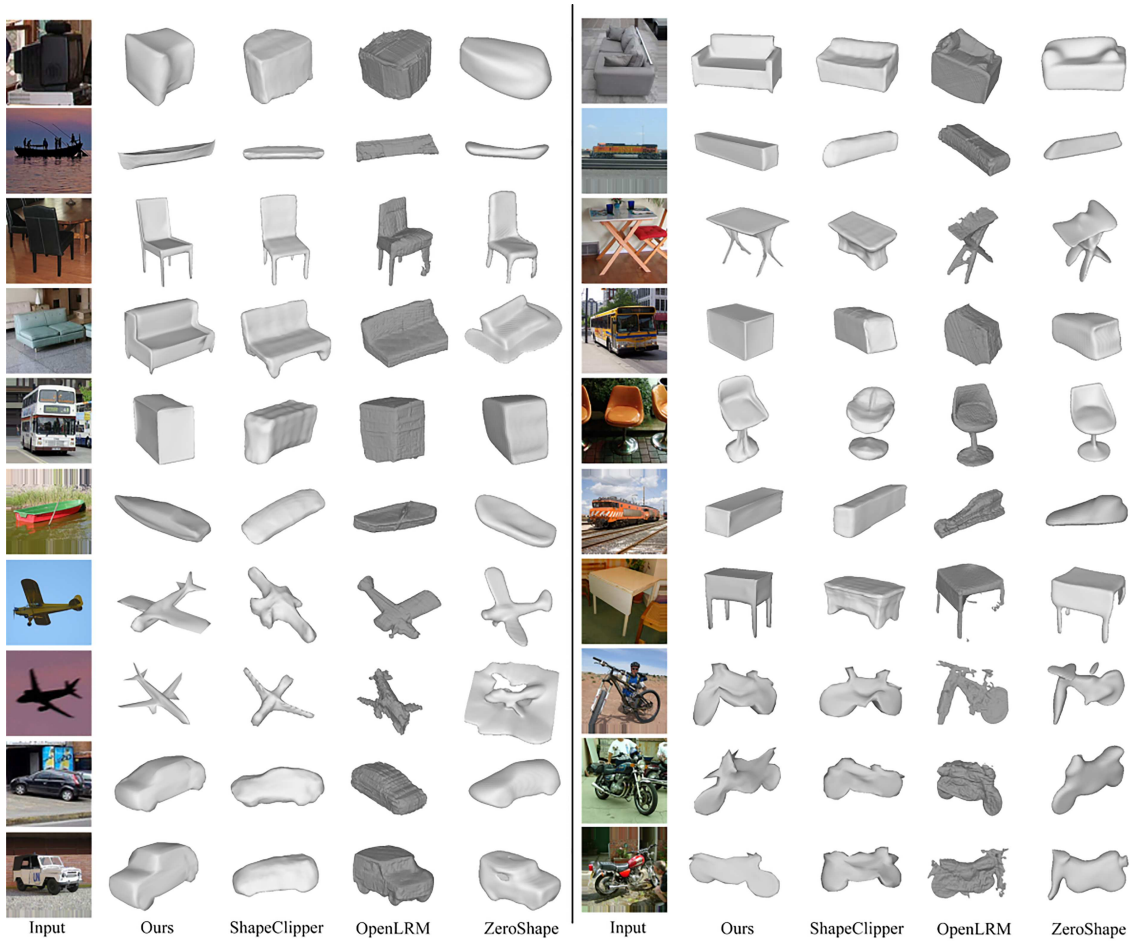

Figure 7 (Color online) Diverse-category visual comparison results on Pascal3D+.

Table 6 Single-category quantitative comparison results on Pascal3D+ Car.

Method	Supervision	3D IoU↑	CD↓
CMR [31]	S+A+C	64.0	–
UMR [38]	S+A	62.0	–
UCMR [37]	S+A	67.3	1.72
MeshInversion [14]	S+P	66.0	–
ShapeClipper [21]	S+A	–	1.82
Unicorn [16]	I	65.9	1.63
Ours	S	67.2	1.47

matched the ground-truth annotations. M-SRN was compared with M-SRN using ground truth mask annotations (M-SRN/GT) and M-SRN using offline segmentation annotations [46] (M-SRN/OF). Table 7 presents the quantitative analysis results. Compared to M-SRN/OF, our foreground perception module effectively improved the reconstruction accuracy, achieving results close to those of M-SRN/GT.

Consistency modules. We confirmed the feature compensation module (FC) and neighbor consistency (NC) module on the ShapeNet dataset. The results of the qualitative and quantitative comparison are presented in Figure 9 and Table 8, respectively. After excluding the FC layer, the object loses details such as wheels and chair legs. When NC is removed, we observed a degraded reconstruction, where the object is only correct when viewed from the input perspective. Additionally, owing to the lack of supervision from new viewpoints, reconstruction often confuses categories, resulting in inaccurate results.

Dynamic neighbor selection. We also validated the effectiveness of the memory-based neighbor selection method on the Pascal3D+ dataset. Two variants of M-SRN were used for comparison: M-SRN using Unicorn neighbor selection (M-SRN/Un) and M-SRN using ShapeClipper's neighbor selection (M-SRN/SC). The results of the quantitative comparisons are presented in Table 9. Because M-SRN/Un relies entirely on the quality of the

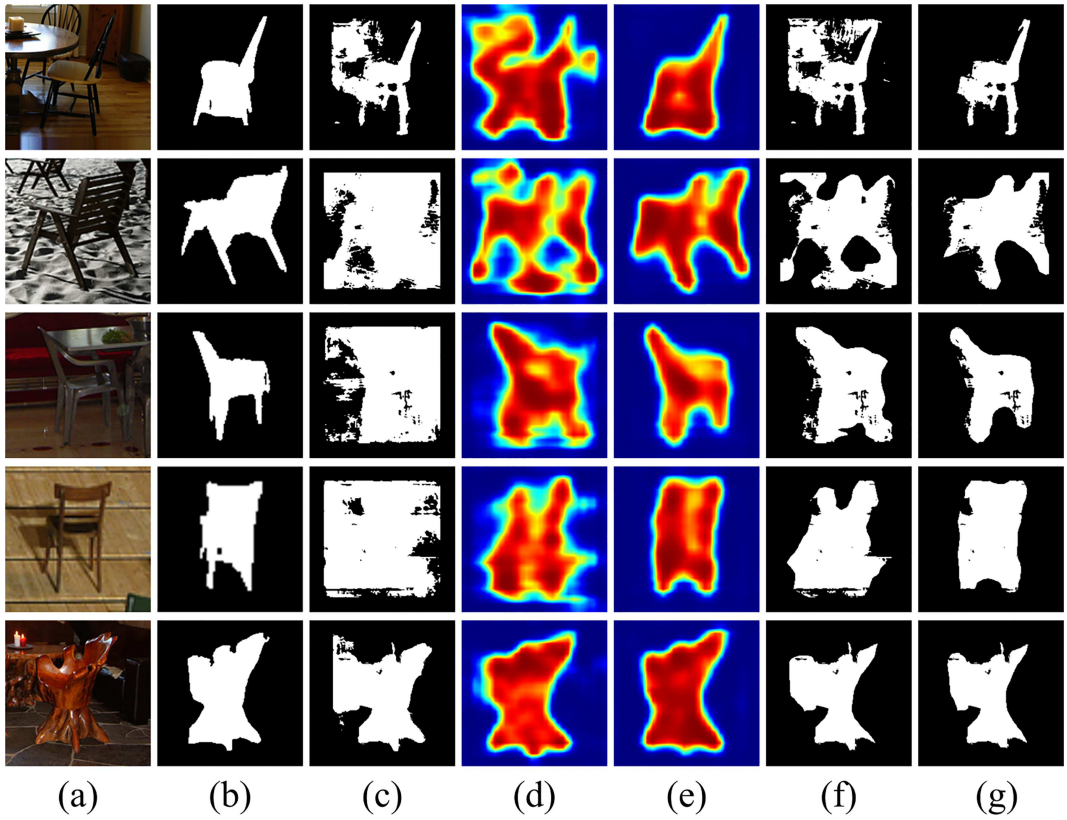

Figure 8 (Color online) Qualitative results of foreground masks. (a) Input images; (b) ground truth masks; (c) offline segmentation masks produced by [46]; (d) activation maps produced by [43]; (e) our activation maps; (f) refined masks using [43]; (g) refined masks using our activation maps.

Table 7 Quantitative ablation results of foreground perception module on Pascal3D+.

	Chamfer distance↓	F-score@0.01↑
M-SRN/OF	3.62	0.164
M-SRN/GT	3.25	0.186
M-SRN	3.41	0.173

Table 8 Quantitative ablation results on ShapeNet.

	Chamfer distance↓	F-score@0.01↑
w/o FC	1.84	0.352
w/o NC	3.81	0.160
M-SRN	1.61	0.365

Table 9 Quantitative results of different neighbor selection methods on Pascal3D+.

	Chamfer distance↓	F-score@0.01↑
M-SRN/Un	4.55	0.145
M-SRN/SC	3.68	0.163
M-SRN	3.41	0.173

encoder, it performs poorly in diverse-category SVR with a wide variety of topological structures. Our dynamic selection method builds on ShapeClipper to further enable the model to eliminate incorrect neighbors during training, thereby achieving higher precision in 3D reconstruction.

4.5.2 *Hyperparameter analysis*

K' represents the number of purified memories for each category in the feature compensation module. We evaluated the performance of the values set to 5, 10, and 20 on the ShapeNet dataset, as listed in Table 10. When K' is set

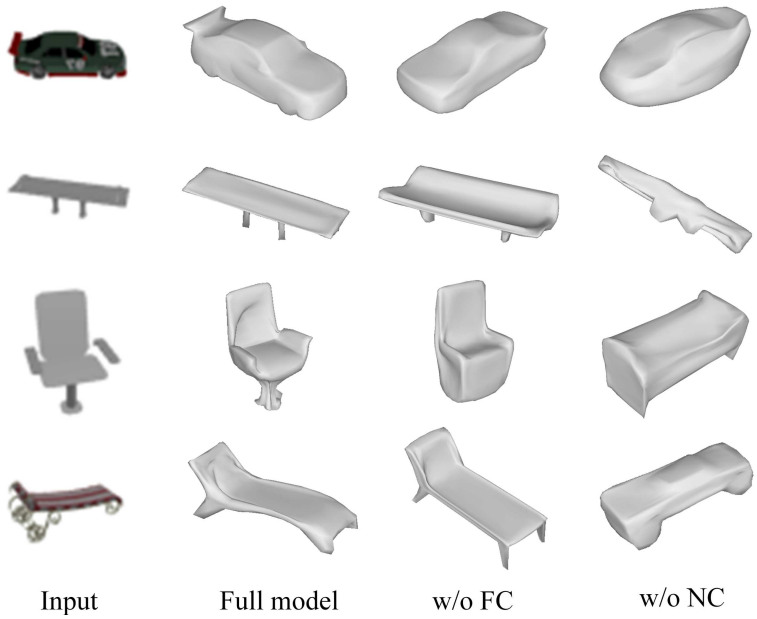

Figure 9 (Color online) Visual ablation results on ShapeNet.

Table 10 Quantitative results of K' on ShapeNet.

	$K'=5$	$K'=10$	$K'=20$
F-score@0.01↑	0.360	0.365	0.363

Table 11 Quantitative results of K'' on Pascal3D+.

	$K''=2$	$K''=5$	$K''=8$
F-score@0.01↑	0.170	0.173	0.166

Table 12 Quantitative results of $k_{\cos}$ on Pascal3D+.

	$k_{\cos}=3$	$k_{\cos}=6$	$k_{\cos}=8$
F-score@0.01↑	0.165	0.173	0.172

to five, M-SRN degrades significantly. K'' represents the number of purified memories in the neighbor consistency enhancement module. As K'' increases, more images are excluded from the preselected neighbor set. We tested the values set to 2, 5, and 8 on Pascal3D+, as listed in Table 11. When K'' is either too low or too high, the model fails to effectively filter neighbors, affecting the accuracy of the results. $k_{\cos}$ represents the number of candidate images for neighbor reconstruction. Unlike SDF-based methods, mesh-based approaches allow us to use more candidate neighbor images. We tested the values of 3, 6, and 8 on Pascal3D+, as listed in Table 12. The performance degraded when $k_{\cos}$ was set to three. When $k_{\cos}$ was greater than six, M-SRN demonstrated stable performance.

5 Conclusion

We proposed a diverse-category single-view 3D reconstruction method called M-SRN. Owing to the utilization of memory representations, the proposed approach achieves dataset-level and category-level consistency reconstructions without requiring mask annotations. The experimental results demonstrate that our method outperforms state-of-the-art weakly supervised methods and achieves results comparable to 3D supervised methods on both synthetic and real-world datasets.

Although M-SRN achieves high-quality diverse category reconstruction, some limitations remain. First, our method struggles with the categories of complex topologies. Second, M-SRN still requires category labels for images, which can be labor-intensive for annotation in large-scale scenarios. Finally, the proposed method can reconstruct a limited number of categories in a single model. We believe that achieving high-quality, weakly supervised zero-shot single-view reconstruction will be an exciting research direction in the future.

Acknowledgements This work was partially supported by Natural Science Foundation of Jilin Province of China (Grant No. 20240101366JC).

References

1 Wang C, Li X, Gu Y F, et al. An adaptive 3D reconstruction method for asymmetric dual-angle multispectral stereo imaging system on UAV platform. Sci China Inf Sci, 2024, 67: 182305
2 Yao Y, Luo Z, Li S, et al. MVSNet: depth inference for unstructured multi-view stereo. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 767–783
3 Zhou Z, Meng M, Zhou Y, et al. Model-guided 3D stitching for augmented virtual environment. Sci China Inf Sci, 2023, 66: 112106
4 Liu C X, Qin J H, Wang S, et al. Accurate RGB-D SLAM in dynamic environments based on dynamic visual feature removal. Sci China Inf Sci, 2022, 65: 202206
5 Fan H Q, Su H, Guibas L, et al. A point set generation network for 3D object reconstruction from a single image. In: Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017. 2463–2471
6 Choy C B, Xu D F, Gwak J Y, et al. 3D-R2N2: a unified approach for single and multi-view 3D object reconstruction. In: Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, 2016. 628–644
7 Wang N, Zhang Y, Li Z, et al. Pixel2Mesh: generating 3D mesh models from single RGB images. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 52–67
8 Yan X C, Yang J M, Yumer E, et al. Perspective transformer nets: learning single-view 3D object reconstruction without 3D supervision. In: Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS), Barcelona, 2016. 1696–1704
9 Tulsiani S, Zhou T H, Efros A A, et al. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017. 209–217
10 Liu S, Li T, Chen W, et al. Soft rasterizer: a differentiable renderer for image-based 3D reasoning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 7708–7717
11 Navaneet K L, Mandikal P, Jampani V, et al. Differ: moving beyond 3D reconstruction with differentiable feature rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019. 18–24
12 Insafutdinov E, Dosovitskiy A. Unsupervised learning of shape and pose with differentiable point clouds. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 2807–2817
13 Duggal S, Pathak D. Topologically-aware deformation fields for single-view 3D reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 1536–1546
14 Zhang J, Ren D, Cai Z, et al. Monocular 3D object reconstruction with GAN inversion. In: Proceedings of European Conference on Computer Vision, 2022. 673–689
15 Navaneet K L, Mathew A, Kashyap S, et al. From image collections to point clouds with self-supervised shape and pose networks. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 1129–1137
16 Monnier T, Fisher M, Efros A A, et al. Share with thy neighbors: single-view reconstruction by cross-instance consistency. In: Proceedings of European Conference on Computer Vision, 2022. 285–303
17 Szymanowicz S, Rupprecht C, Vedaldi A. Splatter image: ultra-fast single-view 3D reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 10208–10217
18 Xu D, Yuan Y, Mardani M, et al. AGG: amortized generative 3D Gaussians for single image to 3D. 2024. ArXiv:2401.04099
19 Huang Z, Stojanov S, Thai A, et al. ZeroShape: regression-based zero-shot shape reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 10061–10071
20 Alwala K V, Gupta A, Tulsiani S. Pre-train, self-train, distill: a simple recipe for supersizing 3D reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 3773–3782
21 Huang Z, Jampani V, Thai A, et al. Shapeclipper: scalable 3D shape learning from single-view images via geometric and clip-based consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 12912–12922
22 Chang A X, Funkhouser T, Guibas L, et al. ShapeNet: an information-rich 3D model repository. 2015. ArXiv:1512.03012
23 Sun X, Wu J, Zhang X, et al. Pix3D: dataset and methods for single-image 3D shape modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 2974–2983
24 Xiang Y, Mottaghi R, Savarese S. Beyond Pascal: a benchmark for 3D object detection in the wild. In: Proceedings of IEEE Winter Conference on Applications of Computer Vision, 2014. 75–82
25 Di Y, Zhang C, Wang P, et al. CCD-3DR: consistent conditioning in diffusion for single-image 3D reconstruction. 2023. ArXiv:2308.07837
26 Melas-Kyriazi L, Rupprecht C, Vedaldi A. PC^2: projection-conditioned point cloud diffusion for single-image 3D reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 12923–12932
27 Kim T, Lee J, Lee K T, et al. Single-view 3D reconstruction based on gradient-applied weighted loss. J Electr Eng Technol, 2024, 19: 4523–4535
28 Tochilkin D, Pankratz D, Liu Z, et al. TripoSR: fast 3D object reconstruction from a single image. 2024. ArXiv:2403.02151
29 Yang X, Lin G, Zhou L. Single-view 3D mesh reconstruction for seen and unseen categories. IEEE Trans Image Process, 2023, 32: 3746–3758
30 Kaiber N E H, Mekhaznia T, Lakhdara Z. Transfer learning-based approach for 3D reconstruction from a single 2D image. In: Proceedings of International Conference on Control, Automation and Diagnosis (ICCAD), 2024. 1–6
31 Kanazawa A, Tulsiani S, Efros A A, et al. Learning category-specific mesh reconstruction from image collections. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 371–386
32 Tulsiani S, Efros A A, Malik J, et al. Multi-view consistency as supervisory signal for learning shape and pose prediction. In: Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, 2018. 2897–2905
33 Lin C H, Wang C, Lucey S. SDF-SRN: learning signed distance 3D object reconstruction from static images. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 11453–11464
34 Hong Y, Zhang K, Gu J, et al. LRM: large reconstruction model for single image to 3D. 2023. ArXiv:2311.04400
35 Liu M, Xu C, Jin H, et al. One-2-3-45: any single image to 3D mesh in 45 seconds without per-shape optimization. In: Proceedings of Advances in Neural Information Processing Systems, 2024
36 Hu T, Wang L, Xu X, et al. Self-supervised 3D mesh reconstruction from single images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 6002–6011
37 Goel S, Kanazawa A, Malik J. Shape and viewpoint without keypoints. In: Proceedings of the 16th European Conference on Computer Vision, Glasgow, 2020. 88–104
38 Li X, Liu S, Kim K, et al. Self-supervised single-view 3D reconstruction via semantic consistency. In: Proceedings of the 16th European Conference on Computer Vision, Glasgow, 2020. 677–693
39 Tulsiani S, Kulkarni N, Gupta A. Implicit mesh reconstruction from unannotated image collections. 2020. ArXiv:2007.08504
40 Ye Y, Tulsiani S, Gupta A. Shelf-supervised mesh prediction in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 8843–8852
41 Zhou B, Khosla A, Lapedriza A, et al. Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2921–2929
42 Xie J, Luo C, Zhu X, et al. Online refinement of low-level feature based activation map for weakly supervised object localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 132–141
43 Wu P, Zhai W, Cao Y. Background activation suppression for weakly supervised object localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 14228–14237

44 Xie J, Xiang J, Chen J, et al. Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation. 2022. ArXiv:2203.13505
45 Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision. In: Proceedings of International Conference on Machine Learning, 2021. 8748–8763
46 Qin X, Zhang Z, Huang C, et al. U2-Net: going deeper with nested U-structure for salient object detection. Pattern Recogn, 2020, 106: 107404
47 Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis, 2015, 115: 211–252
48 Desbrun M, Meyer M, Schröder P, et al. Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 1999. 317–324
49 Nealen A, Igarashi T, Sorkine O, et al. Laplacian mesh optimization. In: Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, 2006. 381–389
50 Kato H, Ushiku Y, Harada T. Neural 3D mesh renderer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 3907–3916
51 Besl P J, McKay N D. A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell, 1992, 14: 239–256
52 Niemeyer M, Mescheder L, Oechsle M, et al. Differentiable volumetric rendering: learning implicit 3D representations without 3D supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 3504–3515