RePre: Improving Self-Supervised Vision Transformer with Reconstructive Pre-training

Luya Wang[1] , **Feng Liang**[2] , **Yangguang Li**[3] , **Honggang Zhang**[1] , **Wanli Ouyang**[4] , **Jing Shao**[3]
[1]Beijing University of Posts and Telecommunications
[2]University of Texas at Austin
[3]SenseTime Group Limited
[4]The University of Sydney
{wangluya,zhhg}@bupt.edu.cn, jeffliang@utexas.edu, {liyangguang,shaojing}@sensetime.com, wanli.ouyang@sydney.edu.au

Abstract

Recently, self-supervised vision transformers have attracted unprecedented attention for their impressive representation learning ability. However, the dominant method, contrastive learning, mainly relies on an instance discrimination pretext task, which learns a global understanding of the image. This paper incorporates local feature learning into self-supervised vision transformers via Reconstructive Pre-training (RePre). Our RePre extends contrastive frameworks by adding a branch for reconstructing raw image pixels in parallel with the existing contrastive objective. RePre is equipped with a lightweight convolution-based decoder that fuses the multi-hierarchy features from the transformer encoder. The multi-hierarchy features provide rich supervisions from low to high semantic information, which are crucial for our RePre. Our RePre brings decent improvements on various contrastive frameworks with different vision transformer architectures. Transfer performance in downstream tasks outperforms supervised pre-training and state-of-the-art (SOTA) self-supervised counterparts.

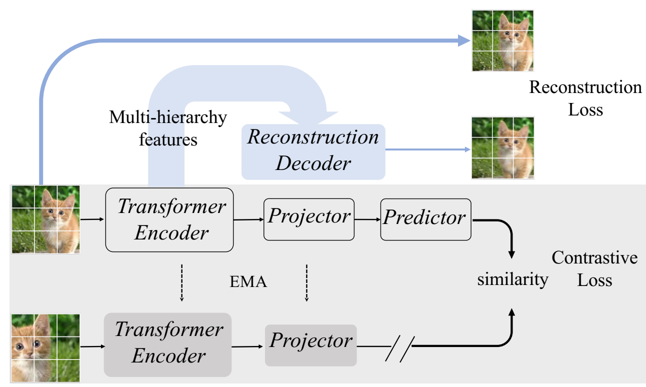

Figure 1: Our RePre extends contrastive frameworks (bottom grey part) by adding a branch for reconstructing raw image pixels (top part). Contrastive framework (MoCo v3 [Chen *et al.*, 2021] in this figure) models image similarity and dissimilarity between two views in an embedding space. Our reconstruction decoder recovers *raw image pixels* using multi-hierarchy features from the transformer encoder.

1 Introduction

Self-supervised pre-training, a method to learn general representations without expensive annotated data, has greatly facilitated Natural Language Processing (NLP) [Radford *et al.*, 2019; Devlin *et al.*, 2018] and similar trends also in Computer Vision (CV) [Chen *et al.*, 2020b; Grill *et al.*, 2020]. One of the main ingredients for the success of self-supervised pre-training in NLP is using the scalable Transformer [Vaswani *et al.*, 2017], a self-attention-based architecture. In CV, Vision Transformer (ViT) [Dosovitskiy *et al.*, 2020] has emerged as an alternative to Convolutional Neural Networks (CNN) since its creation. Despite its remarkable performance, pre-training the vanilla ViT requires enormous labeled data (e.g., JFT-300M [Sun *et al.*, 2017] in [Dosovitskiy *et al.*, 2020]) and extensive computing resources. To avoid expensive labeled data, this paper studies pre-training self-supervised vision transformers.

There is a major difference between the self-supervised pre-training paradigm in NLP and CV: language transformers are pre-trained with masked/autoregressive language modeling [Devlin *et al.*, 2018; Radford *et al.*, 2019], while for vision transformers, the dominant method is contrastive learning which is based on instance discrimination pretext task [Chen *et al.*, 2021; Caron *et al.*, 2021; Xie *et al.*, 2021a]. Concretely, contrastive learning maximizes the similarity of representations obtained from different views of the same image, leading to a global visual understanding (see the bottom part of Fig. 1). However, the sole global feature is insufficient for downstream tasks beyond image classification, such as object detection and segmentation. Motivated by the intuition that a good visual representation should contain global features as well as fine-grained local features, we try to answer: *could we achieve the best of both worlds?*

To achieve holistic visual representations, this paper incorporates fine-grained local feature learning in contrastive self-supervised vision transformers. Inspired by the widely used reconstructive pre-training in CV [Bao *et al.*, 2021], NLP [Devlin *et al.*, 2018], and speech [Hsu *et al.*, 2021], we choose a simple yet effective pretext task: **Re**construction **Pre**-training from raw pixels. Intuitively, pixel reconstructing could let the network capture low semantics to learn fine-grained local features [Ahn and Kwak, 2018]. Our RePre extends contrastive frameworks by adding a branch for reconstructing raw image pixels in parallel with the existing contrastive objective (see Fig. 1). We split an image into patches and *all* these RGB patches are reconstructed through a decoder. Worth mentioning, our neat RePre does not require masking strategy [Hsu *et al.*, 2021; Devlin *et al.*, 2018] nor the tokenizer in BEIT [Bao *et al.*, 2021].

Our initial trial is feeding the output of the last transformer encoder layer into the reconstruction decoder. However, it turns out this simple combination only brings marginal improvements. We argue that this ineffectiveness lies in the discrepancy between the last layer's high semantic features and the low semantic pixel objective. Deep neural networks learn hierarchical semantic features via stacking layers [Dosovitskiy *et al.*, 2020; Liu *et al.*, 2021]. As the processing hierarchy goes up, the early layer captures simple low-level visual information (shallow features), and the late layer can effectively focus on complex high-level visual semantics (deep features). Driven by this analysis, we propose to use the multi-hierarchy features in the transformer encoder. We collect the low to high semantic features within the transformer encoder and use them as a whole to guide the reconstruction.

The reconstruction decoder is another essential part of our RePre. Inspired by U-Net shape [Ronneberger *et al.*, 2015], our decoder gradually integrates the deep to shallow features from multiple hierarchies and regresses to predict the original RGB pixels directly with a simple L_1 loss(see Fig. 2). To combine multi-hierarchy features, the reconstruction decoder is consisted of several fusion layers. Interestingly, we find that the fusion layer could be very *lightweight*, e.g., one or two convolution layers. Since our goal is to introduce additional local features while keeping the high-level semantic features intact, the heavy reconstruction decoder would focus too much on low semantic information, thus harming representation learning. Another favorable property of a lightweight decoder is its little training overload. Our RePre only brings a negligible average of 4% workload in various contrastive frameworks. The reconstruction decoder is only used during pre-training and dropped in the downstream fine-tuning phase.

Our RePre is generic and can be plugged into arbitrary contrastive learning frameworks for various visual translator architectures. Extensive experiments demonstrate the effectiveness and portability of this method. We validate our RePre in the latest contrastive learning frameworks (e.g., DINO, MOCO V3, MoBY, BYOL and SimCLR). Following standard linear evaluation on ImageNet-1K, with RePre, these methods improve top-1 accuracy by 0.5$\sim$1.1%. Prominently, it also brings significant performance to the base methods on dense prediction tasks on the COCO and cityscape datasets, even outperforming supervised methods.

Overall, our contributions are threefold:

1. We incorporate fine-grained local feature learning in contrastive self-supervised vision transformers via adding a reconstruction branch. We adopt a simple yet effective objective: Reconstructive Pre-training (RePre) from raw RGB pixels.
2. RePre utilizes multi-hierarchy fusion to provide rich supervisions from intermediate features. We also find a fast lightweight convolutional reconstruction decoder could bring favorable results.
3. Our RePre is general and easy to be plugged. Decent improvements are observed on various contrastive frameworks with vision transformer and its variants. On dense prediction transfer tasks, RePre also brings significant improvements even outperforming supervised methods.

2 Related Work

2.1 Self-supervised Vision Transformer

Self-supervised contrastive learning has been popular in computer vision. Before the emergence ViT, prior work mainly focus on ResNet, e.g, MoCo [Chen *et al.*, 2020c], SimCLR [Chen *et al.*, 2020b], BYOL [Grill *et al.*, 2020], SimSiam [Chen and He, 2021]. More recently, researchers have incorporated contrastive learning with ViT. MoCo v3 [Chen *et al.*, 2021] proposes an empirical study by training ViT with the MoCo framework. DINO [Caron *et al.*, 2021] shows two new properties of self-supervised ViT compared with supervised ViT. MoBY [Xie *et al.*, 2021a] extends the contrastive framework with a ViT variant, Swin Transformer [Liu *et al.*, 2021]. All these methods share the same spirit: modeling image similarity and dissimilarity (or only similarity) between two or more views, leading to a global image understanding. They lack attention to local and low semantic features, which are crucial for downstream tasks beyond image classification, such as object detection and segmentation. Our RePre is complementary to these contrastive methods via enhancing fine-grained local feature learning.

2.2 Reconstructive Pre-training

Reconstructive (or generative) objectives are highly successful for pre-training in NLP, e.g., masked/autoregressive language modeling in BERT [Devlin *et al.*, 2018] and GPT [Radford *et al.*, 2019]. These methods hold out a portion of the input tokens and train models to predict the missing content. In the field of CV, pioneering iGPT [Chen *et al.*, 2020a] learns a giant self-supervised transformer by directly predicting pixel values, producing competitive results with supervised counterparts. More recently, BEiT [Bao *et al.*, 2021] quantizes image patches as discrete tokens using an off-the-shelf discrete VAE(dVAE) tokenizer [Ramesh *et al.*, 2021], then proposes to predict the masked tokens. Following BEiT, iBoT [Zhou *et al.*, 2021] introduces an online tokenizer. Concurrent MAE [He *et al.*, 2021] and SimMIM [Xie *et al.*, 2021b] propose to reconstruct raw pixels via mask image modeling. Differently, our RePre incorporates reconstructive pixel objectives along with contrastive learning frameworks. It pre-trains general vision transformers for various downstream tasks. Moreover, our neat RePre reconstructs all the image pixels, so it does not require a masking strategy or tokenizer.

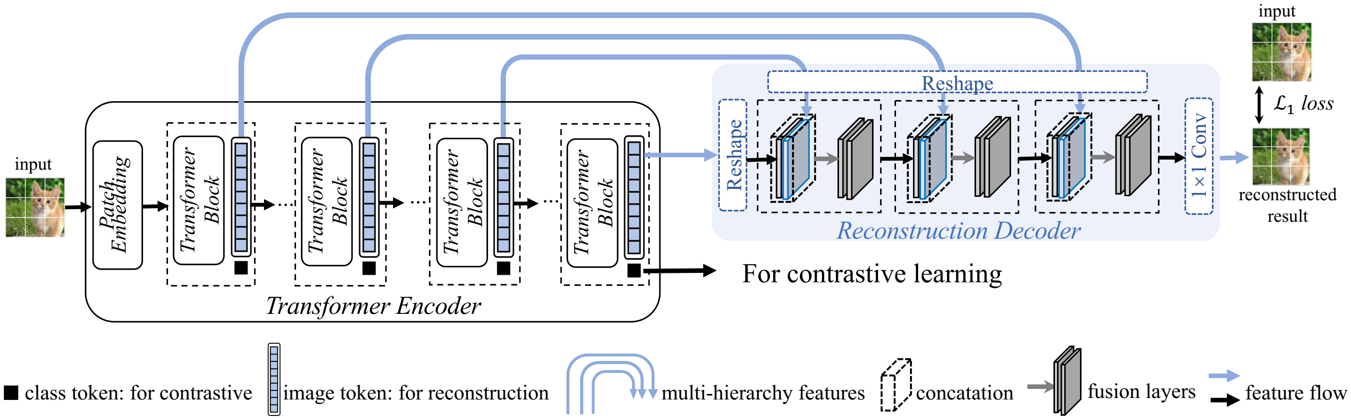

Figure 2: Details of our reconstruction branch. We get low-to-high semantic (multi-hierarchy) features from the transformer encoder by sampling shallow-to-deep transformer blocks. Our decoder gradually integrates deep-to-shallow features and regresses to predict the original RGB pixels with a simple L_1 loss. The sequential image tokens are reshaped to 2D shape for convolution operators in reconstruction decoder. The fusion block in decoder is simple: a concatenation followed by fusion layers. Worth mentioning, for transformer variants with scale downsampling, such as the Swin Transformer, we need to upsample the high-level features before concatenation (details see Sec. 3.3).

3 Method

In this section, we first discuss the contrastive learning frameworks. Then we introduce two key components in our RePre: multi-hierarchy features and a lightweight convolutional decoder (Fig. 2). Finally, we introduce the overall loss function of RePre.

3.1 Revisiting Contrastive Learning Frameworks

The primary focus of contrastive learning is to learn image embeddings that are invariant to different augmented views of the same image while being discriminative among different images. This is typically achieved by maximizing the similarity of representations obtained from different distorted versions of a sample using a variant of Siamese networks. As shown in the bottom part of Fig. 1: a Siamese network is composed of two branches: an online branch and a target branch, where the target branch keeps an Exponential Moving Average (EMA) of the online branch [Chen *et al.*, 2021; Xie *et al.*, 2021a; Caron *et al.*, 2021] or shares weights with the online branch [Chen *et al.*, 2020b; Chen and He, 2021] (not shown in Fig. 1). In particular, each branch encodes an augmented view to a single feature vector in the embedding space, resulting in a level of a global feature.

In order to better prove the scalability and effectiveness of our RePre in arbitrary contrastive learning frameworks, we roughly split current contrastive frameworks into two types: methods with negative samples, e.g., MoCo v3, SimCLR and methods without negative samples, e.g., BYOL, SimSiam.

Methods with negative samples contrast positive samples with negative samples to prevent trivial solutions, i.e., all the outputs collapsing into constant. Specifically, augmented views created from the same samples are considered positive pairs, and images from different samples are considered negative pairs. The target branch outputs the representation of a positive sample and a set of negative samples, and the loss explicitly pulls the pair of positive samples together while pushing apart the pair of negative samples. The loss function can be thought of as a $K+1$ way softmax:

$$\mathcal{L}_{contrast\_w\_neg} = -log\frac{exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} exp(q \cdot k_i/\tau)} \quad (1)$$

Where k_+ is the target feature on the same image as q, called the positive sample of q. k_i is a target feature of negative sample; τ is a temperature term; K is the queue or batch size.

Methods without negative samples only rely on positive samples. They introduce asymmetric architecture to prevent collapse. In particular, it appends a multi-layer perception as predictor to the encoder of the online branch, and it stops the gradient through the target branch. In this case, the loss explicitly pulls together the positive sample pairs, and the objective function is the negative cosine similarity between the two augmented views. Given the output of the online predictor p_1 and the output of the target branch z_2, the objective function is:

$$\mathcal{L}_{contrast\_w/o\_neg} = - < \frac{p_1}{\|p_1\|_2}, \frac{z_2}{\|z_2\|_2} > \quad (2)$$

Where $< \cdot, \cdot >$ denotes the inner product operator.

3.2 Reconstruction with Multi-hierarchy Features

Following the practice of ViT, we split the $H \times W \times 3$ shape image with patch size P. After patch embedding and linear projection, we get $\mathbf{z}_0 \in \mathbb{R}^{(N+1)\times C}$, the sequential feature of an image, where $N = \frac{H}{P} * \frac{W}{P}$. The additional 1 denotes the `class` token, C is the number of channels. The sequential feature would iterate all L transformer blocks within the encoder. We denote the output tokens of each block as $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_L\}$. In contrastive learning, $\mathbf{z}_L^0$ serves as the global image representation. For simplicity, we denote $\mathbf{z}$ excluding $\mathbf{z}^0$ as $\mathbf{y}$, which stands for the representations of patches. For Swin Transformer, because there is no `class` token, we get $\mathbf{y}_0 \in \mathbb{R}^{N\times C}$ after patch embedding. Swin Transformer also has patch merging layers, which reduce the number of token by $\frac{1}{2}$ and increase the feature dimension by

$2\times$. The output embeddings of the last stage are averaged by a global average pooling layer and then sent to a linear classifier for classification, unlike `class` token.

Our initial trial is feeding the output of the last transformer block $\mathbf{y}_L$ into the reconstruction decoder. However, it turns out this simple combination only brings marginal improvements (see Sec. 4.3). We argue that this ineffectiveness lies in the discrepancy between the last layer's high semantic features and the low semantic pixel objective. Inspired by U-Net shape, we collect low-to-high semantic features from shallow-to-deep blocks and reconstruct raw pixels gradually. Given a vanilla ViT with L transformer blocks, we sample K ($K < L$) hierarchical features with even intervals, i.e., our multi-hierarchy feature $\mathbf{Y} = \{\mathbf{y}_{\lfloor \frac{L}{K} \rfloor}, \mathbf{y}_{\lfloor \frac{L}{K} \rfloor * 2}, ..., \mathbf{y}_L\}$, where $\lfloor \cdot \rfloor$ is the floor function. $K = 4$ is a standard practice in this paper. For a $L = 12$ ViT-S, we sample $\mathbf{Y} = \{\mathbf{y}_3, \mathbf{y}_6, \mathbf{y}_9, \mathbf{y}_{12}\}$ as multi-hierarchy feature. For Swin Transformer, which has downsampling operators, we can also get the multi-hierarchy feature. We directly sample the last feature of each resolution stage.

3.3 Lightweight Reconstruction Decoder

With the multi-hierarchy feature, our decoder gradually integrates the deep-to-shallow features and regresses to predict the original RGB pixels directly with a simple L_1 loss(see Fig. 2). Surprisingly, we find that a lightweight convolutional decoder works pretty well (see Sec. 4.3), e.g., one or two fusion layers in each decoder block. A fusion layer consisits a 3×3 convolution layer and a ReLU layer. In order to cooperate with the convolutional operator, the sequential feature $\mathbf{y} \in \mathbb{R}^{N \times C}$ is reshaped to 2D feature $\mathbf{x} \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times C}$. Like in U-Net, the shallow feature is merged into deep feature by concatenation, resulting in a feature with shape $\frac{H}{P} \times \frac{W}{P} \times 2C$. In order to fuse the multi-hierarchy feature, our reconstruction decoder consists of fusion layers in each $K - 1$ block (details in Fig. 2). To predict all pixel values at a full resolution of input images, we apply a 1×1 convolution layer to map each feature vector in the final output of the decoder back to the original resolution. We let this vector take charge of the prediction of corresponding raw pixels. Then, we apply a simple L_1 loss between the original image and the decoder output. In summary, the reconstruction objective is:

$$\mathcal{L}_{reconstruct} = |img - decoder(\mathbf{Y})| \tag{3}$$

Where $|\cdot|$ is theL_1 loss, img is the augmented view before normalization, $\mathbf{Y}$ is the multi-hierarchy feature, $decoder(\cdot)$ returns the reconstructed image.

Our decoder is also compatible with hierarchical vision transformers as Swin Transformer. Because of downsampling, we cannot directly concatenate the deep low-resolution feature with the shallow high-resolution feature. Thus, we apply a bilinear interpolation upsampling operation on the deep feature to make the alignment.

3.4 Overall Loss of RePre

Our RePre is optimized with contrastive loss and reconstructive loss, which simultaneously learns the global features and fine-grained local features. The contrastive loss function is consistent with the contrastive learning method we use (details in Sec. 3.1). The reconstruction loss function computes the mean absolute error between the reconstructed and original images in pixel space (details in Sec. 3.3). We use a weighted sum of these two loss functions as our overall loss. To avoid expensive calculation by optimizing the weights through the grid search method, we incorporate the uncertainty weighting approach proposed by [Kendall *et al.*, 2018]. In particular, each task is weighted by a function of its homoscedastic aleatoric uncertainty rather than by a fixed weight. The overall loss function is as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{contrast} + \lambda_2 \mathcal{L}_{reconstruct} \tag{4}$$

Where λ_1,λ_2 are learnable parameters.

Algorithm 1 Pseudo code of RePre in a PyTorch-like style

```
# reconst_dec: convolutional reconstruction decoder
# online_enc, target_enc: transformer-based encoder
# online_net = online_enc + projector + predictor, predictor is None for symmetric methods
# target_net = target_enc + projector
for x in loader: do
    v1, v2 = aug(x), aug(x) # augmentation
    # # # Reconstructive pre-training # # #
    multi_hierarchy_feats = online_encoder(v1)
    reconst_v = reconst_dec(multi_hierarchy_feats) # reconst_v with shape H * W * 3
    reconstruction_loss = L1_loss(reconst_v, v1) # Details in Sec. 3.3
    # # # Contrastive pre-training # # #
    q1, q2 = online_net(v1), online_net(v2) # [N,C] each
    k1, k2 = target_net(v1), target_net(v2) # [N,C] each
    contrast_loss = ctr_loss(q1, k2) + ctr_loss(q2, k1) # Details in Sec. 3.1
    # # # Combining objectives # # #
    loss = λ1contrast_loss + λ2reconstruction_loss
    loss.backward()
    update(online_net, reconst_dec)
    # target_net is optimized by EMA or gradient
end for
```

4 Experiments

Our RePre is general and can be plugged into arbitrary contrastive learning frameworks with various vision transformer architectures. We first study the linear evaluation of the image recognition task. Then we transfer our pre-trained models into downstream object detection and semantic segmentation tasks. Finally, we do detailed ablation studies about the key components of our RePre.

4.1 Linear Evaluation

Linear evaluation on ImageNet-1K dataset is a standard evaluation protocol to assess the quality of learned representations. After pre-training, we add a linear layer on the top of the network. We only train this linear layer while fixing the pre-trained network.

Fig. 3 and Table 1 list the apparent performance improvements that our RePre brings to different advanced compara-

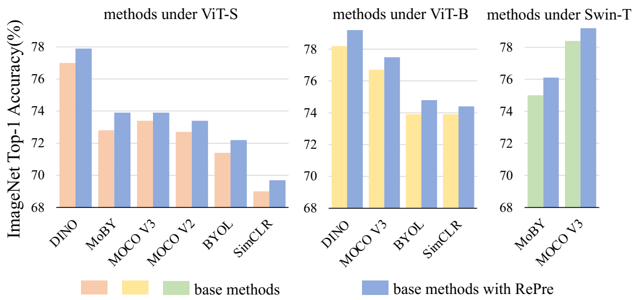

Figure 3: Performance improvements brought by RePre when using different contrastive learning frameworks and network architectures.

Base	Arch.	Epoch	Acc%	Acc w/ RePre
SimCLR		300	69.0	**69.7**(↑ 0.7)
BYOL		300	71.4	**72.2**(↑ 0.8)
MOCO v2		300	71.6	**72.1**(↑ 0.5)
MOCO v2		800	72.7	**73.4**(↑ 0.7)
MOCO v3	ViT-S	300	72.5	**73.2**(↑ 0.7)
MOCO v3		600	73.4	**73.9**(↑ 0.5)
MoBY		300	72.8	**73.9**(↑ 1.1)
DINO		300	75.9	**76.7**(↑ 0.8)
DINO		800	77.0	**77.9**(↑ 0.9)
SimCLR		300	73.9	**74.4**(↑ 0.5)
BYOL		300	73.9	**74.8**(↑ 0.9)
MOCO V3	ViT-B	300	76.5	**77.2**(↑ 0.7)
MOCO V3		600	76.7	**77.5**(↑ 0.8)
DINO		800	78.2	**79.2**(↑ 1.1)
MOCO v3		300	75.4*	**76.4**(↑ 1.0)
MoBY	Swin-T	100	70.9	**71.8**(↑ 0.9)
MoBY		300	75.0	**76.1**(↑ 1.1)

Table 1: More linear evaluation results on ImageNet benchmark. Our RePre brings consistent gains under different methods, architectures and training epochs. * is our reimplementation.

tive learning methods based on different backbone architectures. Our pre-training and fine-tuning recipes are basically the same as the contrastive learning methods. Since our reconstruction decoder is lightweight, our RePre only brings a negligible average of 4% workload. All our experiments are conducted on NVIDIA V100 GPUs.

4.2 Transfer to Downstream Tasks

We further evaluate the transferring performance of the learned representations on downstream tasks of COCO object detection/instance segmentation and Cityscapes semantic segmentation.

Object Detection and Instance Segmentation

We perform object detection/instance segmentation experiments on COCO with Mask R-CNN [He *et al.*, 2017] framework. Following standard practice, we use AdamW optimizer and 1× schedule. The shorter edges of the input images are resized to 800 while the longer side is at most 1333. To compare with advanced research results, we use Swin-T as the backbone. As shown in Table 2, the performance of MoBY with RePre is improved by 1.2% and 0.7% under the same pre-training settings. Similarly, our RePre brings effective performance gains of 0.9% and 1.3% for DINO.

Method	w/ RePre	Epoch	mAP^{bbox}	mAP^{mask}
IN Sup.	–	100	41.6	38.4
	–	300	43.7	39.8
MoBY	×	100	41.5	38.3
	✓	100	**42.1**(↑ 0.6)	**39.2**(↑ 0.8)
	×	300	43.6	39.6
	✓	300	**44.8**(↑ 1.2)	**40.3**(↑ 0.7)
DINO	×	100	42.2	38.7
	✓	100	**43.1**(↑ 0.9)	**40.0**(↑ 1.3)

Table 2: Results of object detection and instance segmentation fine-tuned on MS COCO. We use Mask R-CNN framework with Swin-T as the backbone. Our RePre models outperform supervised ImageNet pre-training and self-supervised DINO with a decent margin.

Method	Arch.	mIoU	mAcc(%)
IN Supervised		71.33	80.30
DINO	ViT-S	72.96	81.32
DINO w/ RePre		**73.40**	**81.95**

Table 3: Results of semantic segmentation fine-tuned on Cityscapes. We use SETR framework with ViT-S as the backbone. All the ViT-S modelsare all pre-trained for 300 epochs.

Semantic Segmentation

We adopt the SETR [Zheng *et al.*, 2021] as the semantic segmentation strategy on Cityscapes and follow the training config as original SETR. For a fair comparison, we use pretrained models based on ViT-S by 300-epoch. As shown in Table 3, DINO with RePre achieves the highest mIoU 73.40% and mAcc 81.95%. It outperforms both supervised and DINO pretrained results. It proves that reconstruction pretraining extracts finer local-level information and is suitable to transfer for semantic segmentation tasks.

4.3 Ablation Study

Two key components of our RePre are multi-hierarchy features and reconstruction decoder. In this part, we perform detailed ablation studies on these two components. Without specification, we use MOCO v3 as the contrastive learning framework and the pre-training epoch is 300.

Ablation on Multi-hierarchy Features

Tab. 4 shows the impact of multi-hierarchy features on performance with the our default convolutional decoder. As we can see, using the 'Single' feature (last layer's output) only brings marginal improvements. We argue that this ineffectiveness lies in the discrepancy between the last layer's high semantic features and the low semantic pixel objective. Using multi-hierarchy features (dubbed 'Multi'), RePre improves MOCO v3 top-1 accuracy performance by 0.7% under DeiT-S and 1.0% under Swin-T. RePre also improves MoBY base-

Method	Arch.	Single	Multi	Top-1 Acc(%)
MOCO v3		–	–	72.5
MOCO v3		✓		72.8
MOCO v3	ViT-S		✓	**73.2**
MoBY		–	–	72.8
MoBY		✓		73.1
MoBY			✓	**73.9**
MOCO v3		–	–	75.4
MOCO v3		✓		75.7
MOCO v3	Swin-T		✓	**76.4**
MoBY		–	–	75.0
MoBY		✓		75.4
MoBY			✓	**76.1**

Table 4: Ablation study on positive impact of multi-hierarchy features. 'Single' and 'Multi' denotes using the last layer output features or using the fused multi-hierarchy features respectively.

Operator	Layer Num.	Arch.	Top-1 Acc(%)
w/o decoder	–		72.5
Conv	1		73.0
Conv	2		**73.2**
Conv	4	ViT-S	73.2
Transformer	1		71.8
Transformer	2		72.0
Transformer	4		71.4
w/o decoder	–		75.4
Conv	1		76.1
Conv	2		**76.4**
Conv	4	Swin-T	76.2
Transformer	1		74.6
Transformer	2		75.2
Transformer	4		74.5

Table 5: Ablation study on the fusion layer in reconstruction decoder. Operator and layer number denotes the type and the number of fusion layers.

line top-1 accuracy performance by 1.1% under DeiT-S or Swin-T. We conjecture that multi-hierarchy features contains different level of semantic information which is crucial for the reconstruction.

We also show the comparison via showing the attention map and t-SNE in Fig. 4. As we can see from the left part, when using multi-hierarchy features, models can identify the edge area of the object more accurately and highlight the core focus. The phoneme can also explain the excellent performance of our RePre when transferring to detection and segmentation tasks. The right part in Fig. 4 shows t-SNE [Van der Maaten and Hinton, 2008] visualization results. Obviously, the learned representations with multi-hierarchy features can be better divided into different categories.

Ablation on Fusion Layer in Reconstruction Decoder

We analyze that convolution can enhance the fine-grained local spatial correlation without damaging context semantics information. We validate it using the same basic Transformer

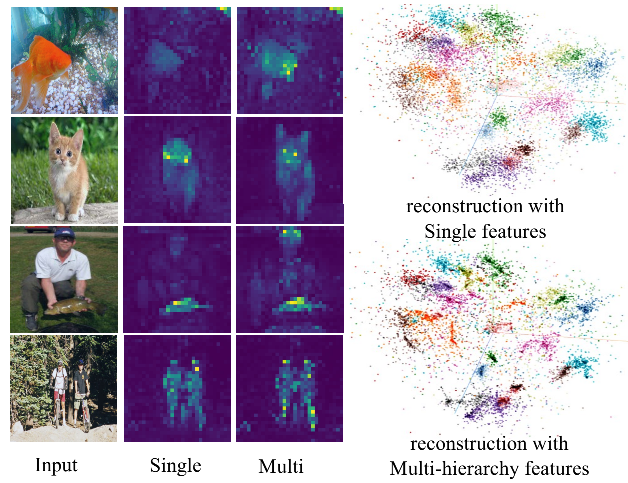

Figure 4: **Left part:** The visualization of attention map. The first column is original images. The second and the third column show `class` token's attentions when using last layer's or multi-hierarchy features.When using multi-hierarchy features, models can identify the edge area of the object more accurately and highlight the core focus. **Right part:** The visualization of t-SNE on ImageNet. We randomly select 20 classes in the validation set. Each point represents embedding from online transformer encoder.

layer as the backbone to replace the 3×3 convolution (Conv) in the decoder fusion layer. Table 5 shows the positive effect brought by convolution. 'Layer Num' represents the repetition times of fusion layer. With a lightweight convolutional decoder, RePre improves baseline top-1 accuracy performance by 0.7% with ViT-S and 1.0% with Swin-T, which could be a strong proof of our hypothesis. The results also validate our analysis that the heavy convoltion or transformer reconstruction decoder would focus too much on low semantic information, thus harming representation learning.

5 Conclusion

This work proposes a simple yet effective objective: Reconstructive Pre-training (RePre) from raw RGB pixels to train self-supervised vision transformers. Our RePre extends contrastive frameworks by adding a branch for reconstructing raw image pixels in parallel with the existing contrastive objective. RePre incorporates local feature learning with a lightweight convolution-based decoder that fuses the multi-hierarchy features from the transformer encoder. Our RePre is generic and can improve arbitrary contrastive learning frameworks with negligible additional cost. With the fact that self-supervised learning in CV was dominated by contrastive objectives in the past several years, we hope our RePre could bring more insights into reconstructive (generative) objectives in this area.

Acknowledgments

This work was funded by National Natural Science Foundation of China under Grant No.62076034. We would like to thank Lan Yang, Kaiyue Pang and Yi-Zhe Song for the help of our work.

References

[Ahn and Kwak, 2018] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proc. of CVPR*, pages 4981–4990, 2018.

[Bao *et al.*, 2021] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[Caron *et al.*, 2021] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint:2104.14294*, 2021.

[Chen and He, 2021] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proc. of CVPR*, pages 15750–15758, 2021.

[Chen *et al.*, 2020a] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proc. of ICML*, pages 1691–1703. PMLR, 2020.

[Chen *et al.*, 2020b] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020.

[Chen *et al.*, 2020c] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint:2003.04297*, 2020.

[Chen *et al.*, 2021] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Grill *et al.*, 2020] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Pires, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. of ICCV*, pages 2961–2969, 2017.

[He *et al.*, 2021] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[Hsu *et al.*, 2021] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *arXiv preprint:2106.07447*, 2021.

[Kendall *et al.*, 2018] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proc. of CVPR*, pages 7482–7491, 2018.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[Ramesh *et al.*, 2021] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015.

[Sun *et al.*, 2017] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proc. of ICCV*, pages 843–852, 2017.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Xie *et al.*, 2021a] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021.

[Xie *et al.*, 2021b] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*, 2021.

[Zheng *et al.*, 2021] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proc. of CVPR*, pages 6881–6890, 2021.

[Zhou *et al.*, 2021] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.