Enhancing Representations through Heterogeneous Self-Supervised Learning

Zhong-Yu Li, Bo-Wen Yin, Yongxiang Liu, Li Liu, Ming-Ming Cheng

Abstract—Incorporating heterogeneous representations from different architectures has facilitated various vision tasks, *e.g.,* some hybrid networks combine transformers and convolutions. However, complementarity between such heterogeneous architectures has not been well exploited in self-supervised learning. Thus, we propose **H**eterogeneous **S**elf-**S**upervised **L**earning (**HSSL**), which enforces a base model to learn from an auxiliary head whose architecture is heterogeneous from the base model. In this process, HSSL endows the base model with new characteristics in a representation learning way without structural changes. To comprehensively understand the HSSL, we conduct experiments on various heterogeneous pairs containing a base model and an auxiliary head. We discover that the representation quality of the base model moves up as their architecture discrepancy grows. This observation motivates us to propose a search strategy that quickly determines the most suitable auxiliary head for a specific base model to learn and several simple but effective methods to enlarge the model discrepancy. The HSSL is compatible with various self-supervised methods, achieving superior performances on various downstream tasks, including image classification, semantic segmentation, instance segmentation, and object detection. The code and dataset are available at https://github.com/NK-JittorCV/Self-Supervised/

Index Terms—self-supervised learning, heterogeneous architecture, representation learning

1 Introduction

Self-supervised learning has succeeded in learning rich representations without requiring expensive annotations. This success is attributed to different pretext tasks, especially instance discrimination [1], [2], [3], [4] and masked image modeling [5], [6], [7]. Adapting these methods to various network architectures, *e.g.,* convolution neural network [8], [9], vision transformer [2], [8], [10], [11] and Swin transformer [12], has brought superior performances on a variety of downstream tasks, including image classification [13], semantic segmentation [14], [15] and object detection [16].

Different neural network architectures learn representations with distinct characteristics that reveal the intrinsic properties of an architecture, *e.g.,* the global and local modeling abilities. Prior works [17], [18], [19], [20] have demonstrated that the characteristics of different architectures can be complementary. Section 1 of the supplementary material also provides a pilot experiment to demonstrate the superiority of combining different architectures over a single architecture. Existing methods [12], [21], [22], [23] mainly focus on architecture design to leverage such complementarity. However, we utilize the complementarity in a representation learning way while not modifying the model architecture.

Inspired by the above analysis, we propose Heterogeneous Self-Supervised Learning (HSSL), which enhances a model with the characteristics of any other architectures. Specifically, during pre-training, the model comprises a base model and an auxiliary head whose architecture is heterogeneous to the base model.

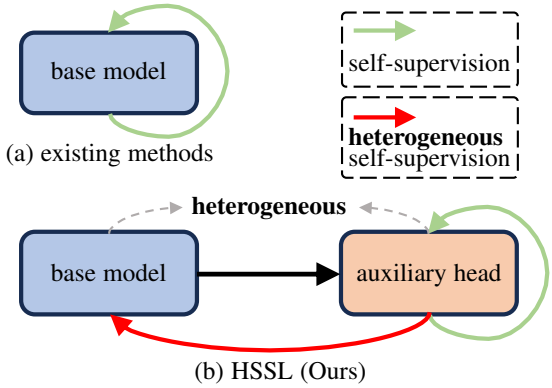


Fig. 1. Illustration of the heterogeneous self-supervised learning (HSSL). (a) General self-supervised learning methods make a base model supervise itself. (b) The HSSL supervises the base model under the guidance of an auxiliary head whose architecture is heterogeneous to the base model, making the base model learn new characteristics.

Such heterogeneity makes the auxiliary head provide missing characteristics from the base model. To endow the base model with its missing characteristics, we encourage the representations of the base model to mimic the representations of the auxiliary head, as shown in Fig. 1. Once pre-training is complete, the base model integrates new characteristics and we remove the auxiliary head.

For a comprehensive analysis, we examine various heterogeneous pairs of the base model and the auxiliary head and discover that the improvement in the base model is positively related to the discrepancy between the base model and the auxiliary head. A more significant discrepancy implies that the auxiliary head can provide more characteristics missing from the base model, thus magnifying the gains of the base model. This observation allows a specific base model to choose the most suitable auxiliary head. We propose a quick search strategy that simultaneously examines all candidate auxiliary heads to perform heterogeneous representation learning with the same base model. Thus, we can

- *Z.-Y. Li, B.-W. Yin, and M.-M. Cheng are with VCIP & TBI Center, Nankai University, Tianjin 300350, China.*
- *M.-M. Cheng is also with NKIARI, Futian, Shenzhen, China.*
- *Y Liu and L Liu are with College of Electronic Science and Technology, National University of Defense Technology (NUDT), Changsha, China.*

- *This work was partially supported by the National Key Research and Development Program of China No. 2021YFB3100800, and the National Natural Science Foundation of China (62225604, 62376283). Computation is supported by the Supercomputing Center of Nankai University.*

quickly determine the most suitable auxiliary head. Moreover, we further modify the chosen auxiliary head to enlarge its discrepancy with the base model to boost the performance.

Our proposed HSSL can be implemented in different self-supervised learning schemes, *e.g.,* contrastive learning [24], self-clustering [2], and masked image modeling [5], thus orthogonal to multiple self-supervised training methods [2], [5], [10], [24]. On various downstream tasks, including image classification [13], semantic segmentation [14], semi-supervised semantic segmentation [25], [26], instance segmentation [16], and object detection [14], [16], HSSL consistently brings significant improvements for various network architectures without structure change.

Our major contributions are summarized as follows:

- We propose heterogeneous self-supervised learning, enabling a base model to learn the characteristics of different architectures.
- Through extensive experiments, we discovered that the discrepancy between the base model and the auxiliary head is positively related to the improvements in the base model and propose a quick search strategy to find the most suitable auxiliary head for a specific base model.
- The proposed representation learning manner is compatible with existing self-supervised methods and consistently boosts performances across various downstream tasks.

2 Related Work

2.1 Self-Supervised Learning

Self-supervised learning enables learning rich representations in the unsupervised setting, reducing the cost of collecting annotations. Early methods design different pretext tasks that can generate free supervision, such as coloration [27], [28], jigsaw puzzles [29], rotation prediction [30], autoencoder [31], [32], image inpainting [33] and counting [34]. The recent success of self-supervised learning can be attributed to instance discrimination [35], [36], [37], [38], [39] and masked image modeling [5], [40], [41], [42], [43] methods. New paradigms, such as correlational image modeling [44] and corrupted image modeling [45], have been proposed, further enriching the field.

Instance discrimination. Instance discrimination generates multiple views of an image through random image augmentations and aligns their representations [46], [47], [48], [49], [50]. This framework has been extended with various loss formulations, including contrastive learning [15], [51], [52], [53], [54], feature alignment [55], [56], [57], clustering assignment [58], [59], [60], [61], [62], redundancy reduction [63], [64], sorting [65], and relational modeling [8], [66]. These methods have been applied at both image-level [52], [67], [68], [69] and dense-level [70], [71], [72], [73], and demonstrate broad adaptability across architectures, including convolutional neural networks [9], vision transformers [2], and Swin Transformers [12]. However, existing approaches often overlook the potential complementarity between different architectures. In this work, we propose HSSL, a framework designed to harness complementary characteristics across different architectures using a heterogeneous self-supervised learning scheme. Moreover, our method is orthogonal to existing self-supervised techniques.

Masked image modeling. The masked image modeling (MIM) based methods [74], [75], [76], [77] reconstruct masked image patches based on the unmasked ones, emphasizing spatial context learning. Researchers have explored diverse reconstruction targets to capture representations with varying properties. For example, pixel-based reconstruction [5], [41], [78], [79], [80], [81] often yields strong yet non-linear representations. To endow representations with strong semantic information, more types of targets, *e.g.,* hand-designed HOG [82], [83], [84], frequency [80], [85], masked positions [86], features from online network [66], [87], [88], [89], [90], discretized tokens [40], [91], [92], or the combination of multiple targets [91], [93], [94]. Recent research [66] also reconstructs representations from an off-the-shelf pre-trained model and achieves excellent performance, especially when using large-scale datasets [95]. When using the online network [90], [96], some works [6], [10], [97], [98], [99], [100] further combine the advantages of masked image modeling and instance discrimination to boost the performance. Meanwhile, apart from targets, some works [74], [77], [79], [101] also investigate different masking strategies to facilitate high-level representations.

Similar to instance discrimination, masked image modeling [10], [102], [103], [104], [105], [106], [107], [108] has also been applied to diverse architectures like vision transformer [109], ConvNext-V2 [102], and Swin [12]. These developments underscore the potential of leveraging architectural diversity to improve MIM-based representation learning.

2.2 Heterogeneity on Neural Network

The heterogeneous neural network, which combines multiple types of architectures [20], [23], [114], can generate complementary characteristics and facilitate various vision tasks, including semantic segmentation [17], [115], [116], object detection [117], [118], image classification [18], [119], [120], and image quality assessment [121]. These methods mainly design new architectures to leverage complementarity. For example, Wu *et al.* [120] combines convolution and attention in an architecture to achieve better classification accuracy. In comparison, we enforce a network constructed by a specific architecture to learn characteristics from any other architectures via representation learning without any structural changes. Thus, the proposed method is flexible in fusing characteristics from any architectures.

Some works [19], [122] have tried to utilize the complementarity to improve self-supervised learning. Specifically, these methods make the ViT and ResNet guide each other. However, beyond this pair, they lack a comprehensive analysis and understanding of the complementarity between different architectures. In comparison, we investigate a wide range of architectures, not only ViT and ResNet, and provide a comprehensive analysis of why and how the complementarity benefits self-supervised learning. We discover that a more significant model discrepancy leads to more significant improvements, enabling us to design more suitable auxiliary heads to guide a specific model.

3 Method

In Section 3.1, we recall the existing self-supervised methods. Then, in Section 3.2, we describe the proposed heterogeneous self-supervised learning and demonstrate its compatibility with existing methods. In Section 3.3, we demonstrate that the improvements come from the complementarity of heterogeneous architectures. Section 3.4 analyzes what makes a good auxiliary head and discovers that a greater model discrepancy brings more benefits. Inspired by this discovery, we propose a quick

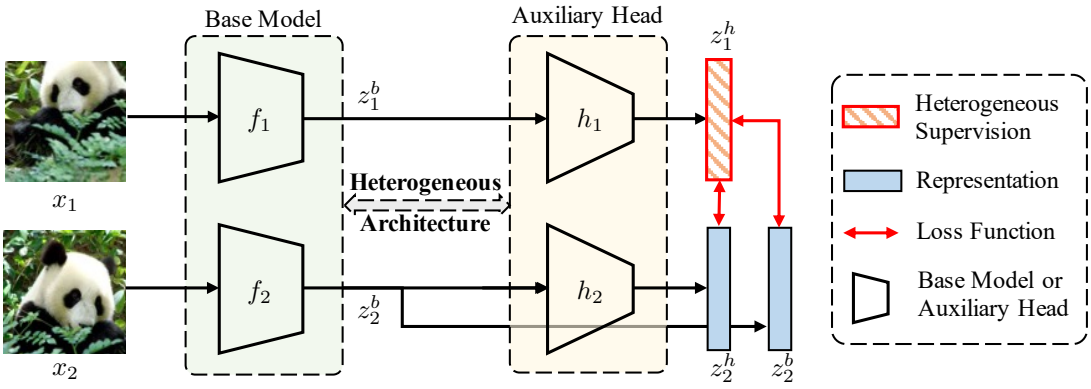


Fig. 2. Our HSSL framework. The architectures of the base model and the auxiliary head are heterogeneous. The representations extracted by the auxiliary head supervise the two networks simultaneously. The base model and the auxiliary head can be arbitrary architectures, such as ViT [109], Swin [12], ConvNext [110], ResNet [111], ResMLP [112], and PoolFormer [113].

search strategy to choose the most suitable auxiliary head for a specific base model in Section 3.5 and several simple but effective methods that enlarge the model discrepancy to bring more improvements in Section 3.6.

3.1 Preliminaries

The HSSL can be implemented in different forms, *e.g.,* instance discrimination and masked image modeling. In this paper, we mainly use the instance discrimination framework as the illustrative example. We first briefly recall the common framework of instance discrimination. Given an image x, different views of x, *i.e.,* x_1 and x_2, are generated by different data augmentations. Their representations, *i.e.,* z_1 and z_2, are extracted by teacher and student networks, respectively. Then, instance discrimination maximizes the similarity between z_1 and z_2. Specifically, the loss function has different forms [2], [10], [123], and we abstract the loss as $\mathcal{L}(z_1, z_2)$.

3.2 Heterogeneous Supervision

Denoting the backbone used by existing methods [2], [10] as the base model, HSSL utilizes an auxiliary head, whose architecture differs from the base model, to endow the base model with its missing characteristics. The overall pipeline is visualized in Fig. 2. For simplification, we refer to the base model/auxiliary head at the teacher and student branches as f_1/h_1 and f_2/h_2, respectively. Given x_1 and x_2, the base models extract representations $z_1^b = f_1(x_1)$ and $z_2^b = f_2(x_2)$. Then, the auxiliary head takes these representations as input and output $z_1^h = h_1(z_1^b)$ and $z_2^h = h_2(z_2^b)$. Since heterogeneous architectures extract z_1^h/z_2^h and z_1^b/z_2^b, the z_1^h/z_2^h contains a part of the characteristics that are missing from the z_1^b/z_2^b. The base model can learn those missing characteristics with the loss function $\mathcal{L}(z_1^h, z_2^b)$, which pulls z_1^h and z_2^b together.

Meanwhile, to guarantee that the auxiliary head can learn meaningful characteristics, we also pull representations extracted by auxiliary heads in teacher and student together, *i.e.,* using the loss function $\mathcal{L}(z_1^h, z_2^h)$. The base model and the auxiliary head are pre-trained simultaneously, and the total loss function $\mathcal{L}$ can be defined as follows:

$$\mathcal{L} = \mathcal{L}(z_1^h, z_2^b) + \mathcal{L}(z_1^h, z_2^h). \tag{1}$$

During pre-training, the auxiliary head is serially connected at the end of the base model, enabling the former to learn meaningful characteristics with only a few layers. Thus, the increased training time and memory costs are negligible. After pre-training, we remove the auxiliary head and only reserve the base model.

Incorporating HSSL into different SSL methods. The proposed HSSL is compatible with different self-supervised learning (SSL) methods, including MoCo [24], DINO [2], iBOT [10], and MAE [5], as shown in Tab. 6. When combined with different methods, the loss function defined in Equ. (1) takes on distinct forms. For clustering based methods [2], [10], the representations are transformed into probability distributions over K dimensions through some projection heads and a softmax function, and the loss function is defined as follows:

$$\mathcal{L} = -\sum_{i=1}^{K}(z_1^h)_i \log((z_2^b)_i) - \sum_{i=1}^{K}(z_1^h)_i \log((z_2^h)_i), \tag{2}$$

where the projection heads and the softmax function are hidden for simplification. Additionally, other forms of loss functions can also be combined with HSSL, *e.g.,* InfoNCE [124] in contrastive learning [2] and reconstruction loss in masked image modeling [5]. For more details, please refer to Section 6 of the supplementary material.

Analysis for various architectures. To validate the effectiveness of the proposed HSSL, we evaluate the impact of different auxiliary heads on the base model. In this analysis, we aim to explore the effect of diverse architectures on the HSSL. Thus, we choose ResNet [111], PoolFormer [113], ResMLP [112], ConvNext [110], ViT [109], and Swin [12] as the auxiliary heads due to their diverse architectures. For example, ResNet [111] is a classic convolutional network based on local convolutions, and ConvNext [110] further adopts large kernel convolutions. ViT [109] is a transformer network based on global self-attention, and Swin [12] integrates local attention in the transformer architecture. Moreover, PoolFormer [113] and ResMLP [112] adopt different modeling mechanisms beyond convolutional and transformer architectures, *i.e.,* pooling and spatial MLP. As shown in Tab. 1, using the auxiliary head can consistently enhance the base model across all pairs[1]. Furthermore, we observe that an auxiliary head that is heterogeneous to the base model brings more gains than a homogeneous one. For example, when using

1. For all experiments in Section 3 and Section 5, we adopt the ImageNet-S_{300} dataset [25], which contains 300 categories from ImageNet-1K [13], to save computational costs.

TABLE 1
Effects of various auxiliary heads on different base models.

		Base Model			
		ViT		ResNet	
		Top-1	Top-5	Top-1	Top-5
	Baseline	67.5	84.4	63.2	84.3
Auxiliary Head	ViT [109]	68.0	84.7	64.0	84.3
	Swin [12]	69.4	85.9	63.9	84.4
	PoolFormer [113]	70.1	86.3	63.9	84.5
	ResNet [111]	71.7	86.9	63.5	84.3
	ResMLP [112]	72.6	87.8	64.4	84.9
	ConvNext [110]	72.7	87.6	63.7	84.4

TABLE 2
Weak auxiliary heads also enhance strong base models. Experiments without a declared auxiliary head mean the baselines of corresponding base models.

Base model	Auxiliary head	Top-1
ViT [109]	-	67.5
ViT [109]	ResNet [111]	71.7
ViT [109]	ResMLP [112]	72.6
ResMLP [112]	-	58.0
ResMLP [112]	ViT [109]	59.6
Swin [12]	-	72.8
Swin [12]	PoolFormer [113]	73.7
Swin [12]	ResMLP [112]	73.4

ViT as the base model, the auxiliary head of the ViT only improves by 0.5% in Top-1 accuracy. In comparison, the auxiliary head of the ConvNext brings a 4.2% improvement in Top-1 accuracy. These results and observations prove the necessity of the proposed HSSL method. We also investigate whether relatively weaker auxiliary heads can enhance stronger base models. Tab. 2 shows positive results. For example, the weaker PoolFormer [113] improves the Top-1 accuracy of the Swin [12] base model by 0.9%. This indicates that our HSSL method is robust to different model architectures and can brings consistent improvements under different settings.

3.3 Heterogeneity Brings Gains

While the HSSL takes effect across different pairs of the base model and the auxiliary head, we further explore how the auxiliary head enhances the base model. Specifically, we observe that the auxiliary head can solve a part of samples that the base model cannot. To illustrate this, we first define sets B_1, B_2, and H, which contain the samples that can be correctly solved by the base model pre-trained by baseline (DINO [2]), the base model pre-trained by HSSL, and the auxiliary head pre-trained by HSSL, respectively. Meanwhile, U means the set that contains all samples of a dataset. Then $H \cap (U - B_1)$ contains the samples that the auxiliary head can solve but are beyond the capacity of the base model pre-trained by baseline. The number of these samples is defined as follows:

$$N_s = |H \cap (U - B_1)|. \tag{3}$$

Taking ViT as the base model, we show that the auxiliary head can solve some samples that are beyond the ability of the base model in Tab. 3. More importantly, an auxiliary head, which can solve more samples unsolved by the base model, brings more significant improvements to the base model.

TABLE 3
Auxiliary head solves samples that the base model (ViT) cannot solve. The sIoU and N_s represent the degree of overlap between HSSL and the original version on the correct samples and the number of newly added correct samples from HSSL. Their details are provided in Section 3.3.

Auxiliary Head	Top-1	N_s	sIoU
ViT [109]	68.0	792	59.5
Swin [12]	69.4	854	60.7
PoolFormer [113]	70.1	904	60.8
ResNet [111]	71.7	1061	67.8
ResMLP [112]	72.6	1270	72.9
ConvNext [110]	72.7	1278	70.2

We further investigate whether the base model can address those samples in $H \cap (U - B_1)$ under the guide of the auxiliary head. After pre-training by HSSL, both the base model and the auxiliary head can address some samples that are beyond the capacity of the baseline. These samples can be represented as $B_2 \cap (U - B_1)$ and $H \cap (U - B_1)$ for the base model and the auxiliary head, respectively. We notice that there exists a substantial overlap between these two subsets. The degree of overlap can be quantified as follows:

$$\text{sIoU} = \frac{|B_2 \cap (U - B_1) \cap H \cap (U - B_1)|}{|B_2 \cap (U - B_1)|}. \tag{4}$$

Tab. 3 shows the sIoU obtained by different auxiliary heads when using ViT as the base model. For example, there is a 70% overlap when using ConvNext [110] as the auxiliary head. The high overlap demonstrates that the improvements in the base model can mainly be attributed to complementarity and heterogeneity.

3.4 Analysis of Model Discrepancy

Different auxiliary heads produce different effects for a specific base model, as shown in Tab. 1. For ViT, a transformer-based base model, using ConvNext as the auxiliary head is more suitable than the others. When ResNet is the base model, utilizing ResMLP and ViT as auxiliary heads can complement global modeling ability and bring more significant improvements. The above observation motivates us to delve deep into what makes a good auxiliary head. By investigating different architectures, we discover that a more significant discrepancy between the base model and the auxiliary head brings more gains to the base model. This phenomenon inspires us to propose a search strategy to quickly determine the most suitable auxiliary head for a specific base model in Section 3.5 and several simple but effective methods to magnify the discrepancy in Section 3.6.

Model discrepancy. During heterogeneous self-supervised learning, the auxiliary head learns a part of characteristics that are missing from the base model itself. That is to say, there exists a representation discrepancy between the base model and the auxiliary head, *i.e.,* the discrepancy between z_1^b and z_1^h. Taking the self-clustering based methods [2], [10] as an example, the z_1^b defined in Section 3.1 means probability distributions over K dimensions. Then, we use the Kullback-Leibler divergence to measure the discrepancy as follows:

$$\mathcal{D} = -(z_1^b)^T \log(\frac{z_1^h}{z_1^b}), \tag{5}$$

where z_1^b and z_1^h are extracted from the teacher network after pre-training.

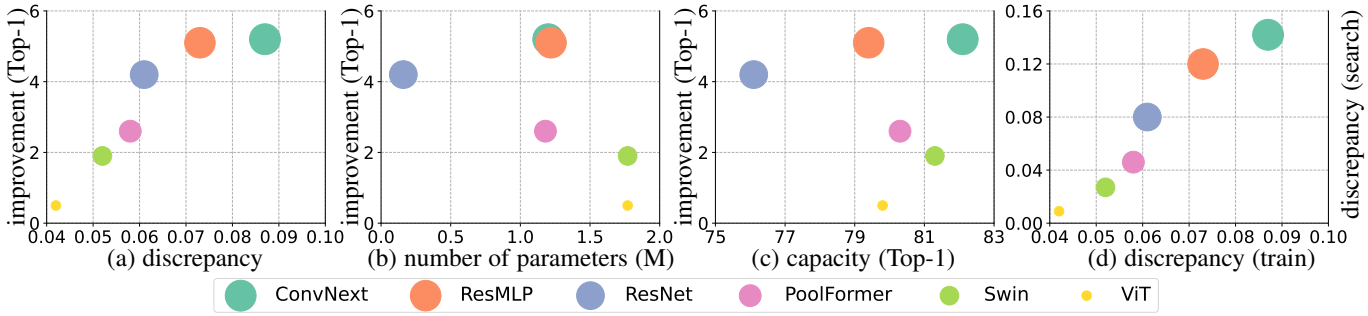


Fig. 3. In (a)-(c), we visualize the relationship between the improvements in the base model (ViT-S/16) and three factors, including (a) the representation discrepancy between the base model and the auxiliary head, (b) the number of parameters of a 1-layer auxiliary head, (c) The capacity of the architecture that is used to build the auxiliary head. For the capacity of each architecture, we use the supervised classification accuracy on ImageNet-1K, reported in the official paper of each architecture, as a reference to its capacity. In (d), we show a consistent trend between the discrepancies obtained by searching and examining each auxiliary head individually. In all figures, the size of the dot is positively related to the improvement brought by the corresponding auxiliary head.

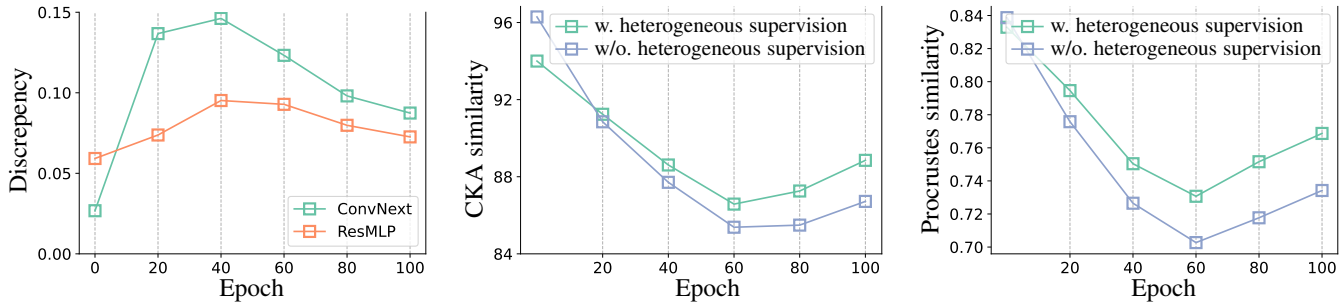


Fig. 4. Training dynamics of the discrepancy or similarity between the base model and the auxiliary head during pre-training. Left: The discrepancy $\mathcal{D}$ (defined in Equ. (5)) between the base model and the auxiliary head when using ConvNext [110] or ResMLP [112] as the auxiliary head and using ViT [109] as the base model. Middle: The feature-level CKA similarity between the base model and the auxiliary head. Right: The feature-level Procrustes similarity between the base model and the auxiliary head.

More significant discrepancy leads to greater improvements. Taking ViT-S/16 as an example of the base model, in Fig. 3 (a), we show its improvement when it learns from each auxiliary head and its discrepancy with each auxiliary head. It can be observed that there is a positive relationship between improvements and discrepancies. A more significant discrepancy means the auxiliary head learns more characteristics that are missing from the base model, thus prompting the base model to complement more characteristics.

To further confirm whether the improvement comes from the heterogeneity, we analyze other factors, including the number of parameters of the auxiliary head and the capacity of the architecture used to build the auxiliary head, where we use the supervised classification accuracy on ImageNet-1K [13], which is reported by the official paper of each architecture, as a reference to the architecture capacity. As shown in Fig. 3 (c) and (d), both factors have no positive correlation with the improvement. For example, ViT [109] has a larger capacity than ResNet [111], but ResNet is more suitable than ViT when serving as the auxiliary head. These results demonstrate that a greater improvement is not from a stronger auxiliary head but the heterogeneity.

The dynamic of model discrepancy. Based on the discrepancy analysis, we investigate how auxiliary heads influence the base model during pre-training. Fig. 4 provides detailed insights into the interaction between the base model and auxiliary heads, showing the discrepancy $\mathcal{D}$ (as defined in Equ. (5)), CKA similarity, and Procrustes similarity between the base model and the auxiliary head, respectively. From Fig. 4, we observe that the discrepancy initially increases and then decreases during training. Notably, heterogeneous supervision significantly amplifies the discrepancy and reduces the similarity, as evident in the middle and right panels. This observation suggests that heterogeneous supervision encourages the base model to learn from the auxiliary head. Moreover, the left panel demonstrates that using ConvNext as the auxiliary head induces a larger discrepancy than ResMLP when ViT is employed as the base model. This aligns with previous analysis, which indicates that a larger discrepancy can lead to greater performance improvements. Naturally, the model discrepancy provides us with the possibility to select the optimal auxiliary head for a specific base model.

3.5 Searching for Suitable Auxiliary Heads

A suitable auxiliary head provides more characteristics missing from a specific base model, thus complementing the base model better and producing higher improvements. However, in the unsupervised setting, there is no annotated data to evaluate each auxiliary head. Inspired by the positive relationship between the discrepancies and improvements, we use the model discrepancy to determine the most suitable auxiliary head for a specific base model via a label-free approach. However, due to the vast number of candidate auxiliary heads, testing candidates one by one is time-consuming. Thus, we propose an efficient search strategy to find the auxiliary head with the largest discrepancy to the base model through one quick training.

Quick Search Strategy. Unlike the standard HSSL architecture, which employs a single auxiliary head, we arrange all the

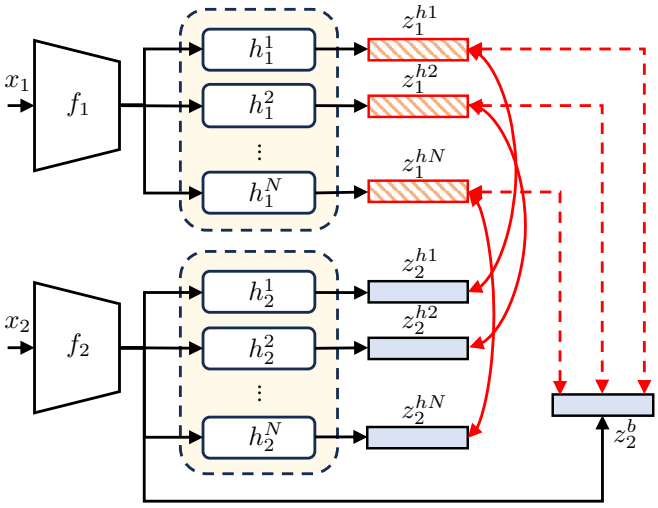


Fig. 5. Illustration of the quick search strategy. Given N distinct architectures, we construct N different auxiliary heads, where $h^i_{1/2}$ represents the auxiliary head built using i-th architecture. The subscripts 1 and 2 indicate teacher and student branches, respectively. In the figure, the red dotted lines and solid lines correspond to the loss of the first and second terms of Equ. (6). Projection heads are omitted from for clarity.

TABLE 4
Cooperation of multiple auxiliary heads when using ViT as the base model. '$\mathcal{D}$' represents the discrepancy degree between the auxiliary head and the base model.

Auxiliary Head	$\mathcal{D}$	Top-1	Top-5
ResMLP	7.3e-2	72.6	87.8
ConvNext	8.7e-2	72.7	87.6
ConvNext+ResMLP	11.0e-2	73.7	88.2

candidate auxiliary heads in parallel during training, as shown in Fig. 5. This allows each auxiliary head to independently perform heterogeneous self-supervised learning without interference. Suppose there are N candidate auxiliary heads, each corresponding to a distinct architecture. For the inputs x_1 and x_2, we first send them to the base models in teacher and student branches to generate representations $z_1^b = f_1(x_1)$ and $z_2^b = f_2(x_2)$, respectively. Then, in the teacher branch, the N auxiliary heads further process z_1^b and produce heterogeneous representations $\{z_1^{hi} \mid i \in [0, N-1]\}$. Similarly, the student branch generates $\{z_2^{hi} \mid i \in [0, N-1]\}$. For the i-th auxiliary head, we define the loss function like Equ. (1) as follows:

$$\mathcal{L}^{hi} = \mathcal{L}(z_1^{hi}, z_2^b) + \mathcal{L}(z_1^{hi}, z_2^{hi}). \tag{6}$$

The overall loss function across all auxiliary heads is given by:

$$\mathcal{L}^s = \frac{1}{N}\sum_{i=0}^{N-1} \mathcal{L}^{hi}. \tag{7}$$

In practice, the features in Equ. (6) are passed through independent projection heads before calculating the loss, as commonly adopted in prior work [2], [5]. During searching, we apply an independent projection head for each auxiliary head. To minimize mutual interference, the representations $z^b_{1/2}$ of the base model are passed into separate projection heads when paired with different auxiliary heads. For clarity, these projection heads are omitted in Equ. (6) and Equ. (7).

TABLE 5
Analysis of the shortcut connection in the auxiliary head. Here, we adopt ViT as the base model and ConvNeXt as the auxiliary head.

The first shortcut	$\mathcal{D}$	Top-1	Top-5
Preservation	5.6e-2	71.0	86.8
Removal	8.7e-2	72.7	87.6

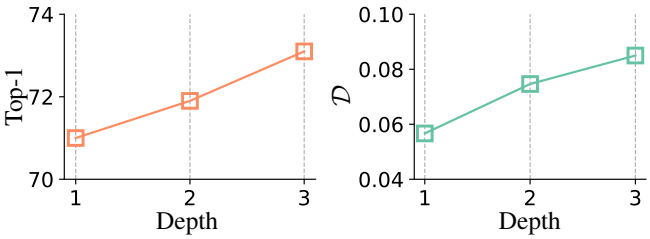


Fig. 6. Influence of network depth in the auxiliary head. We take the ViT as the base model and ConvNext as the auxiliary head.

After training with Equ. (7), we calculate the discrepancy between the base model and each auxiliary head. For the i-th head, the discrepancy $\mathcal{D}_i$ is computed as:

$$\mathcal{D}_i = -(z_1^b)^T \log(\frac{z_1^{hi}}{z_1^b}). \tag{8}$$

Finally, the auxiliary head with the largest discrepancy is selected:

$$\arg\max_i \mathcal{D}_i, \tag{9}$$

where the i-th auxiliary head is identified as the most complementary auxiliary head to the base model. Therefore, the optimal auxiliary head for the base model can be rapidly identified.

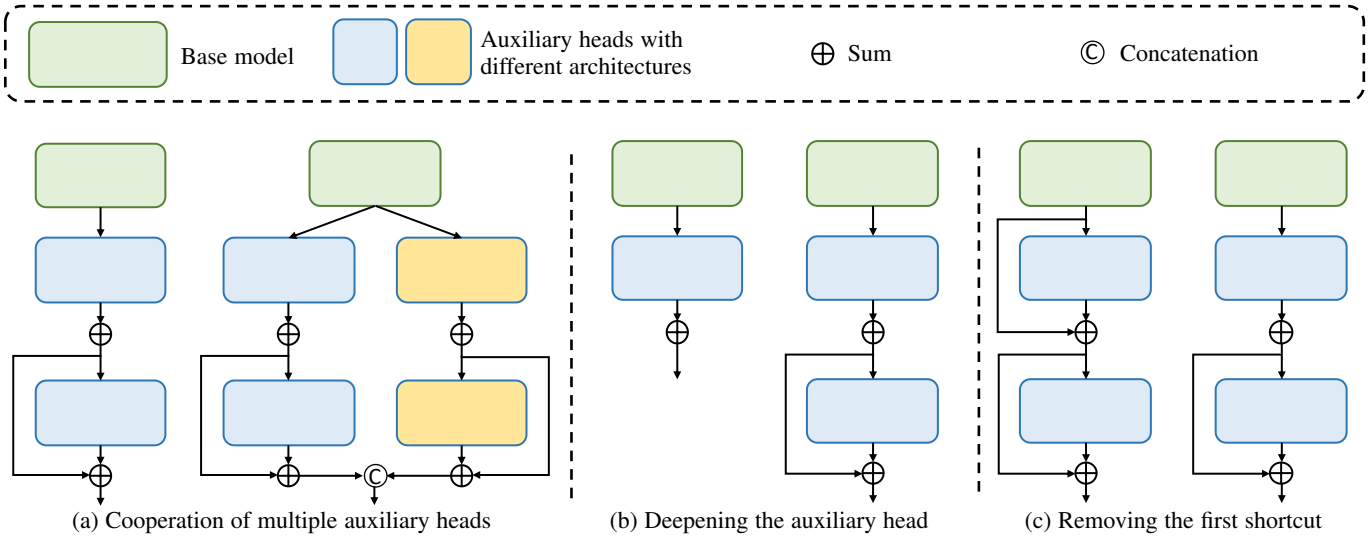
Searching time. Compared to examining each auxiliary head through multiple training, the proposed search strategy requires only one training. Because we use a very shallow auxiliary head, the base model accounts for most of the computational budget during training. As a result, when there are six auxiliary heads, training with all of them simultaneously, *i.e.,* the proposed search strategy, requires only 1.4× training time than training with one. Thus, the search strategy requires only about $\frac{1.4\times1}{1\times6} \approx 23\%$ of the time required by examining all auxiliary heads one by one. Meanwhile, we empirically discover that using only 10% of the training data is enough for searching, further reducing the search time significantly.

Searching results. Taking ViT as the base model, we analyze the relative relationship of its discrepancies with different auxiliary heads. As shown in Fig. 3 (b), the relative relationship obtained during searching aligns with that obtained by testing each auxiliary head individually, verifying the effectiveness of the search strategy.

3.6 Enlarging the Model Discrepancy

Section 3.4 demonstrates that a larger model discrepancy can bring more improvement gains. Inspired by this observation, we propose three simple but effective technologies to magnify such discrepancy and further boost the performance.

Cooperation of multiple auxiliary heads. The base model only learns limited characteristics from a specific auxiliary head.



(a) Cooperation of multiple auxiliary heads (b) Deepening the auxiliary head (c) Removing the first shortcut

Fig. 7. Three strategies to enlarge the model discrepancy. For each strategy, the left shows the baseline and the right shows the specific strategy.

Inspired by the principles of ensemble learning, where combining multiple models can bring performance improvements, we try to combine multiple auxiliary heads built by distinct architectures to complement more characteristics that are missing from the base model. Specifically, supposing there are n auxiliary heads composed of different architectures, we represent them as $\{h_1^i | i \in [1,n]\}$ and $\{h_2^i | i \in [1,n]\}$ in teacher and student, respectively. With the representations $z_{1/2}^b$ produced by the base model, each auxiliary head $h_{1/2}^i$ processes them and generates representations $h_{1/2}^i(z_{1/2}^b)$. As shown in Fig. 7 (a), these representations are combined as follows:

$$z_{1/2}^{hc} = \text{concat}(\{h_{1/2}^i(z_{1/2}^b) | i \in [1,n]\}), \tag{10}$$

where concat means the concatenation along the channel dimension. Then, we substitute these representations into Equ. (1) and get the new loss functions as follows:

$$\mathcal{L} = \mathcal{L}(z_1^{hc}, z_2^b) + \mathcal{L}(z_1^{hc}, z_2^{hc}). \tag{11}$$

Compared to a single auxiliary head, multiple ones can provide more characteristics required by the base model. As shown in Tab. 4, when using two auxiliary heads simultaneously, *i.e.,* ConvNext [110] and ResMLP [112], we achieve greater improvements than just using ConvNext or ResMLP. This demonstrates that our HSSL method can achieve further improvements through cooperation of multiple auxiliary heads.

Deepening the auxiliary head. Several works [125], [126] have observed that representations learned across different layers of a model exhibit discrepancies, with larger gaps between layers resulting in greater disparities. Meanwhile, a deeper network can learn more powerful representations [111]. Thus, we are inspired to deepen the auxiliary head to make it learn more characteristics different from the base model. This is implemented by simply stacking more blocks on the auxiliary head, as shown in Fig. 7 (b). The results in Fig. 6 verify that deepening the auxiliary head from one to three layers enlarges the discrepancy and brings greater improvements.

Removing the first shortcut connection. In most architectures, the shortcut connection is utilized by default to ensure convergence. However, in our HSSL, we observe that the shortcut in the first layer of the auxiliary head reduces the discrepancy between the base model and the auxiliary head. To illustrate this argument, we take a two-layer auxiliary head as an example, where the two layers are represented as F_1 and F_2, respectively. We use z to represent the output of the base model. When remaining the first shortcut, the auxiliary head outputs $z+F_1(z)+F_2(z+F_1(z))$. In comparison, the auxiliary head outputs $F_1(z)+F_2(F_1(z))$ when we remove the first shortcut. We can observe that the former directly adds z, *i.e.,* the output of the base model, to the output of the auxiliary head, thus reducing the model discrepancy. This phenomenon is further illustrated in the supplementary material. Thus, we remove the first shortcut connection, as shown in Fig. 7 (c). This approach enlarges the discrepancy and leads to more significant improvements, as shown in Tab. 5.

4 Experiments

4.1 Experimental Settings

We integrate HSSL with a wide range of self-supervised methods, including MoCov3 [24], DINO [2], AttMask [74], iBOT [10], MAE [5] and MFF [81]. For each method, we follow its official implementation. During pre-training, we adopt ViT-S/16 or ViT-B/16 architecture as the base model, and the auxiliary head uses the ConvNext architecture unless otherwise specified. In the auxiliary head, we default the depth to 3 and remove the shortcut connection at the first layer. More details about pre-training and fine-tuning are shown in the supplementary material.

4.2 Experimental Results

Image classification on ImageNet-1K. We first fully fine-tune the base models on ImageNet-1K and compare the classification performance, as shown in Tab. 6. Using ViT-B/16, HSSL improves by 0.5% in Top-1 accuracy over iBOT [10] when pre-training for 400 epochs. With 600 epochs, HSSL can achieve 84.1% Top-1 accuracy, even outperforming iBOT of 1600 epochs. We also combine HSSL with masked image modeling (MIM) based methods, and the implementation details are shown in the supplementary material. In Tab. 6, HSSL enhances MAE by 0.4%

2. For a fair comparison, we report effective epochs [10] that account for actual images used during pre-training. For iBOT and DINO that use 10 local crops, the number of effective epochs is four times the number of actual epochs.

TABLE 6
Cooperating the proposed HSSL with various architectures and frameworks. We report Top-1 on the validation set of ImageNet-1K [13], using the evaluation protocols of fully fine-tuning, linear probing, and k-NN, respectively. † means that we use the multi-crop strategy [2] with 2 global crops of 224×224 and 10 local crops of 96×96.

	Architecture	Epoch[2]	Fine-tuning	Linear	k-NN
MAE [5]	ViT-S/16	400	80.4	-	-
MAE [5]+HSSL			80.8	-	-
MoCo [24]	ViT-S/16	200	-	65.3	57.4
MoCo [24]+HSSL			-	65.7	58.1
DINO [2]	ViT-S/16	200	-	67.9	61.2
DINO [2]+HSSL			-	70.8	65.5
iBOT [10]	ViT-S/16	200	-	71.3	65.2
iBOT [10]+HSSL			-	72.6	67.3
DINO [2]†	ViT-S/16	400	-	74.6	70.9
DINO [2]†+HSSL			-	75.7	72.5
iBOT [10]†	ViT-S/16	400	80.9	74.4	71.5
iBOT [10]†+HSSL			81.3	76.5	72.8
AttMask [74]†	ViT-S/16	400	-	76.1	72.8
AttMask [74]†+HSSL			-	76.7	73.1
iBOT [10]†	ViT-B/16	400	83.3	77.8	74.0
iBOT [10]†+HSSL			83.8	79.4	75.3
iBOT [10]†	ViT-B/16	1600	84.0	79.5	77.1
iBOT [10]†+HSSL		600	84.1	79.6	76.0
MFF [81]	ViT-B/16	300	83.3	-	-
MFF [81]+HSSL			83.6	-	-
DINO [2]	Swin-T	200	-	69.2	60.1
DINO [2]+HSSL			-	71.2	65.1
DINO [2]	PVT-Small	200	-	67.7	61.3
DINO [2]+HSSL			-	69.6	64.4

on Top-1 accuracy after pre-training ViT-S/16 for 400 epochs. Compared to MFF [81], we advance the performance by 0.3% Top-1 accuracy after pre-training ViT-B/16 for 300 epochs.

We also evaluate the effectiveness of HSSL using k-NN and linear probing on the ImageNet-1K dataset. As shown in Tab. 6, HSSL consistently improves various methods, including instance discrimination based (*e.g.,* DINO [2] and MoCo [24]), and hybrid methods that combine instance discrimination with MIM (*e.g.,* iBOT [10] and AttMask [74]). For example, when pre-training ViT-B/16 by 400 epochs, HSSL advances iBOT by 1.6% in linear probing accuracy. Meanwhile, HSSL can achieve comparative performances over iBOT with even fewer epochs (600 vs. 1600 epochs). These results show that HSSL enhances the ability of classification and is orthogonal to existing representation learning methods. Moreover, Tab. 6 highlights that HSSL can enhance different transformer architectures, extending beyond the plain vision transformer [109]. For example, HSSL improves the Swin-T [12] and PVT-Small [133] by 2.0% and 1.9% in linear probing accuracy, respectively, after pre-training for 200 epochs.

Finally, we directly compare our HSSL with prior methods, as shown in Tab. 7. Compared to instance discrimination based methods, our method pre-trained for 600 epochs achieves 84.1% and 79.6% Top-1 accuracy on fully fine-tuning and linear probing, respectively, outperforming methods such as iBOT [10] that requires 1600 epochs to achieve 79.5% linear probing accuracy. Even with a shorter pre-training schedule of 400 epochs, HSSL still surpasses iBOT by 0.5% and 1.6% on fully fine-tuning and linear probing, respectively. Although some MIM-based methods, *e.g.,* ccMIM [101] and BootMAE [94] deliver competitive performance in fine-tuning, they lag in linear probing accuracy and may exhibit limited effectiveness on downstream tasks, as shown in Tab. 8. In contrast, HSSL achieves superior performance across both linear probing, fine-tuning, and downstream tasks.

TABLE 7
Comparison with previous methods using ViT-B/16 [109]. † means the usage of a pre-trained perceptual codebook for the tokenization.

	Epochs[2]	Fine-tuning	Linear
MoCo [24]	600	83.2	76.7
DINO [2]	1600	83.6	78.2
SimMIM [41]	800	83.8	56.7
MAE [5]	1600	83.6	68.0
iBOT [10]	400	83.3	77.8
MaskFeat [83]	1600	84.0	-
BootMAE [94]	800	84.2	66.1
SdAE [90]	300	84.1	64.9
BEiT [40]	800	83.2	56.7
SiameseIM [88]	400	83.7	76.8
MOKD [122]	400	-	78.0
LocalMIM [84]	1600	84.0	-
MFF [81]	800	83.6	-
CIM [45]	300	83.3	-
ConMIM [87]	800	83.7	39.3
ccMIM [101]	300	83.6	66.9
ccMIM [101]	800	84.2	68.9
PeCo [92]†	300	84.1	-
SERE [8]	400	83.7	77.9
iBOT [10]+HSSL	400	83.8	79.4
iBOT [10]+HSSL	600	84.1	79.6
MFF [81]+HSSL	300	83.6	-

TABLE 8
Comparison on semantic segmentation. We fine-tune UperNet [127] with the ViT-B/16 [109] as the backbone on the ADE20K [128] dataset, following existing works [5], [10].

	Architecture	Epochs[2]	mIoU
MoCo [24]	ViT-B/16	600	47.2
DINO [2]		1600	46.8
MAE [5]		1600	48.1
BootMAE [94]		800	49.1
SdAE [90]		300	48.6
BEiT [40]‡		800	45.6
SiameseIM [88]		400	49.6
MixedAE [129]		800	48.7
LocalMIM [84]		1600	49.5
MFF [81]		800	48.6
ConMIM [87]		800	46.0
ccMIM [101]		800	47.7
PeCo [92]†		300	48.5
SERE [8]		800	50.0
iBOT [10]	ViT-B/16	400	47.9
iBOT [10]		1600	50.0
iBOT [10]+HSSL		400	50.3
iBOT [10]	ViT-S/16	400	45.2
iBOT [10]		3200	45.4
iBOT [10]+HSSL		400	46.1

Transfer learning on image classification. Besides ImageNet-1K, we also transfer the pre-trained base models to other classification datasets, including CIFAR [131] and iNaturalist [132]. As shown in Tab. 11, HSSL brings consistent improvements across different datasets, demonstrating superior transferability.

TABLE 9
Comparison on object detection and instance segmentation with ViT-B/16. We fine-tune the models on the COCO [16] dataset and report AP^m as segmentation mask AP and AP^b as bounding box AP, respectively.

	Architecture	Epochs[2]	AP^m	AP^b
DINO [2]	ViT-B/16	1600	43.4	50.1
MAE [5]		1600	44.3	51.3
SERE [8]		400	43.8	50.7
MFF [81]	ViT-B/16	300	43.2	50.0
MFF [81]+HSSL		300	43.7	50.5
iBOT [10]	ViT-B/16	400	43.2	50.1
iBOT [10]+HSSL		400	44.0	51.0
iBOT [10]	ViT-B/16	1600	44.2	51.2
iBOT [10]+HSSL		600	44.3	51.4

TABLE 10
Cross-domain transferring learning on RAW object detection. The models are fine-tuned on AODRaw [130], which collects RAW images for object detection. Apart from the Average Precision (AP) [16], we also report AP_{75} and AP_{50} at the IoU threshold of 0.75 and 0.50. AP_s, AP_m, and AP_l mean the AP for small, medium, and large objects.

	Architecture	Epochs[2]	AP	AP_{50}	AP_{75}
DINO [2]	Swin-T [12]	200	28.9	45.7	30.2
DINO [2]+HSSL		200	29.5	46.5	30.8

TABLE 11
Transfer learning on more image classification benchmarks, including CIFAR [131] and iNaturalist [132].

	Architecture	Epochs[2]	$Cifar_{100}$	$INat_{18}$	$INat_{19}$
iBOT [10]	ViT-B/16	400	92.1	74.0	78.4
iBOT [10]		1600	92.2	74.6	79.6
iBOT [10]+HSSL		400	92.2	75.2	79.7

Transfer learning on semantic segmentation. We use UperNet [127] as the segmentation model for evaluating semantic segmentation performance. Following prior works [10], we fine-tune the models on the ADE20K [128] dataset. As shown in Tab. 8, HSSL achieves 50.3% mIoU after pre-training ViT-B/16 for 400 epochs. Notably, HSSL outperforms iBOT [10], which requires 1600 epochs of pre-training to achieve similar results. Using ViT-S/16 pre-trained for 400 epochs, HSSL also advances iBOT [10] by 0.9% mIoU. These results highlight the effectiveness of HSSL on dense prediction.

Transfer learning on instance segmentation. Following [10], we use Cascade Mask R-CNN [134] to implement instance segmentation and object detection. As shown in Tab. 9, HSSL advances iBOT [10] by 0.8% AP^m and 0.9% AP^b with just 400 epochs of pre-training. Additionally, compared to MFF [81], HSSL delivers a 0.5% improvement in both AP^m and AP^b. Notably, HSSL also reduces training costs, achieving superior performance by lowering the required pre-training epochs from 1600 (as in iBOT [10]) to just 600.

Cross-domain transferring. We also evaluate the pre-trained models on the AODRaw [130] dataset, designed for object detection in the RAW domain. The RAW domain presents a significant domain gap compared to the sRGB domain on which we pre-train models. The results show that our HSSL achieves a 0.6% improvement in AP when pre-training Swin-T for 200 epochs, highlighting its strong cross-domain generalization capability.

TABLE 12
Semi-supervised classification on ImageNet-1K [13]. We utilize linear and k-NN classifiers with 1%/10% labels and report the Top-1 accuracy.

	Architecture	Epochs[2]	1%	10%
iBOT [10]	ViT-B/16	400	64.8	76.3
iBOT [10]+HSSL		400	66.1	76.8

TABLE 13
Semi-supervised semantic segmentation on ImageNet-S [25]. We report the mIoU on the val and test sets. The PT means pre-trained weights initiate the model, and FT means fully fine-tuned weights initiate the model, respectively.

	Architecture	Epochs[2]	$ImageNet\text{-}S_{PT}$		$ImageNet\text{-}S_{FT}$	
			val	test	val	test
iBOT [10]	ViT-B/16	400	48.3	47.8	62.6	63.0
iBOT [10]		1600	50.5	50.1	-	-
iBOT [10]+HSSL		400	51.5	51.1	63.5	63.1

Semi-supervised learning. Collecting annotations requires huge costs. Semi-supervised learning can reduce the demand for expensive annotations. Thus, we also evaluate the ability of HSSL in semi-supervised classification and semantic segmentation. We follow the paradigm in [10] for semi-supervised classification to fine-tune the pre-trained base models with a part of labels. As shown in Tab. 12, HSSL improves by 1.3% and 0.5% in Top-1 accuracy over iBOT [10] when using 1% and 10% training labels, respectively. For semi-supervised semantic segmentation, we fine-tune the base models on the ImageNet-S [25] dataset, in which 919 categories and 9190 labeled images are included. Tab. 13 reports the mIoU on the val and test sets. We can observe that HSSL significantly improves iBOT [10] by 4.7% and 4.2% mIoU on the val and test sets.

Unsupervised semantic segmentation. We evaluate the pre-trained base models with unsupervised semantic segmentation. For training, we follow the pipeline proposed in [25] and consider three datasets [25], *i.e.,* $ImageNet\text{-}S_{50}$, $ImageNet\text{-}S_{300}$, and ImageNet-S datasets. As shown in Tab. 14, HSSL outperforms iBOT by 1.8% mIoU on the ImageNet-S dataset. The results in semi-supervised and unsupervised learning show that HSSL benefits the perception and recognition in the absence of labels.

Time and memory usage. Tab. 15 shows the time and memory usage required by iBOT [10] and our HSSL. Compared to the baseline, the HSSL only increases negligible computation costs because the serial connection between the base model and the auxiliary head enables the auxiliary head to extract helpful representations with just a few layers.

5 Ablation and Analysis

We perform ablation studies by pre-training models for 100 actual epochs on the $ImageNet\text{-}S_{300}$ to save computation costs. By default, we set the depth of the auxiliary head to 1. We evaluate the performance by reporting knn classification accuracy (Cls.) on the ImageNet and segmentation mIoU (Seg.) on the ImageNet-S.

TABLE 14
Unsupervised semantic segmentation on ImageNet-S [25]. 919/300/50 mean the ImageNet-S/ImageNet-S_{300}/ImageNet-S_{50} datasets, respectively. We follow the pipeline and setting proposed in [25] and report mIoU on the val and test sets. Here, we do not adopt the multi-crop strategy for the representation learning.

	Datasets	Architecture	Epochs[2]	val	test
iBOT [10]	50	ViT-S/16	400	46.2	45.1
iBOT [10]+HSSL				54.4	54.5
iBOT [10]	300	ViT-S/16	200	22.2	22.4
iBOT [10]+HSSL				26.6	26.0
iBOT [10]	919	ViT-S/16	200	12.2	11.3
iBOT [10]+HSSL				14.0	13.6

TABLE 15
Time and memory usage during pre-training on an 8-GPU machine, with a batch size of 256 and 10 multi-crops of 96×96.

	Architecture	Epochs[2]	Time (h)	Memory (G)
iBOT [10]	ViT-B/16	400	82.7	18.3
iBOT [10]+HSSL			94.5	21.4

TABLE 16
Ablation for the supervision manner on the base model. **B** and **A** mean the base model and the auxiliary head, respectively. **A**→**B** means that the auxiliary head supervises the base model. **B**→**B** means the base model supervises itself.

	Seg.	Cls.	
	mIoU	Top-1	Top-5
A→A	16.1	26.5	48.0
A→A + B→B	31.4	68.0	86.4
A→A + **A**→**B**	36.9	72.7	87.6

TABLE 17
Ablation for the structure of the auxiliary head.

	Seg.	Cls.	
	mIoU	Top-1	Top-5
MLP	35.8	70.0	86.3
Token Mixer	36.3	70.1	86.4
MLP + Token Mixer	36.9	72.7	87.6

TABLE 18
Ablation for the shared projection and not shared projection.

Shared proj.	Seg.	Cls.	
	mIoU	Top-1	Top-5
✔	35.8	72.3	87.5
✘	36.9	72.7	87.6

Effect of the supervision manner. After connecting the auxiliary head, we investigate whether to use the base model itself or the auxiliary head to guide the base model, where the former is homogeneous and the latter is heterogeneous. As shown in Tab. 16, the heterogeneous manner outperforms the homogeneous manner, achieving 5.5% higher mIoU and 4.7% higher Top-1 accuracy. These results verify that heterogeneous supervision is essential, allowing the base model to learn complementary characteristics from the auxiliary head.

Structure of the auxiliary head. We use a unified framework for different auxiliary heads, which includes a token mixer and an MLP block. Here, we take ConvNext as an example and evaluate the effect of the token mixer and MLP block. The results presented in Tab. 17 show that the token mixer plays a more crucial role, leading to improvements of 0.6% mIoU and 2.6% Top-1 accuracy compared to the MLP block.

TABLE 19
Ablation for parallel and serial connections of the auxiliary head. We use a depth of 3 for serial connection. We show the multiples relative to the baseline for the time and memory costs.

	Seg.	Cls.		Computation cost	
	mIoU	Top-1	Top-5	time	memory
baseline	29.3	67.5	84.4	×1.00	×1.00
parallel	34.6	72.8	87.2	×2.53	×2.25
serial	37.1	73.9	88.4	×1.09	×1.12

TABLE 20
Utilizing heterogeneous self-supervision on different granularity when using ViT, taking iBOT [10] as an example.

Image-level	Patch-level	Seg.	Cls.	
		mIoU	Top-1	Top-5
✘	✘	42.3	75.1	89.3
✔	✘	46.2	75.8	89.4
✔	✔	46.7	76.0	89.5

TABLE 21
Comparison with the strategy of deep-to-shallow (DTS) [25].

	Seg.	Cls.	
	mIoU	Top-1	Top-5
baseline	29.3	67.5	84.4
+DTS	30.5	68.6	85.4
+HSSL	36.9	72.7	87.6

Whether to share the projections. Before calculating the losses, self-supervised learning methods usually process the teacher/student representations through some projection heads. Tab. 18 investigates whether to share the projections between the base model and the auxiliary head. The results indicate that not sharing the projections provides an advantage of 1.1% mIoU and 0.4% Top-1 accuracy. Due to the different architectures, the representations between the base model and the auxiliary head have discrepancies, and not sharing the projections allows greater flexibility in processing the discrepancy.

Parallel or serial connection for the auxiliary head. We can connect the auxiliary head and the base model in serial or parallel. For the parallel connection, we use the entire ConvNext-Tiny as the auxiliary head that directly takes the images as input. In contrast, the serial connection allows the auxiliary head to extract rich information with just a few layers. As shown in Tab. 19, the parallel arrangement requires about 2.32 × training time compared to the serial connection. Moreover, using serial connection achieves better performances than parallel arrangement, achieving better computational efficiency.

Cls token and patch token. Some methods [8], [10], [74] calculate losses on different granularity simultaneously. Taking iBOT [10], which considers losses on both image-level and patch-level,

as an example, Tab. 20 shows the effects when using HSSL on different granularity. Note that when only using HSSL on the image-level, the pixel-level self-supervision is only used between the base models of teacher and student. It can be seen that the base model can learn the majority of the helpful information from the auxiliary via only image-level supervision. Meanwhile, learning with pixel-level supervision also brings further improvement. These results show that we can save computational costs by only applying HSSL on the image-level.

Comparison with deep-to-shallow. The deep-to-shallow enhances the representations of a shallow layer with supervision from a deeper layer within a homogeneous architecture. As shown in Tab. 21, this strategy only leads to a slight improvement in the ViT, likely because the deep and shallow layers in ViT are highly similar [135], making the supervision lack diversity. In contrast, the heterogeneous self-supervised learning prompts the ViT to learn diverse knowledge, achieving significant improvements of 6.4% mIoU and 4.1% Top-1 accuracy over the DTS.

6 Conclusion

In this paper, we propose heterogeneous self-supervised learning (HSSL). Specifically, we enforce a base model to learn from an auxiliary head whose architecture is heterogeneous to the base model, endowing the base model with some characteristics that are missing from itself. Furthermore, we discover that the discrepancy between the base model and the auxiliary head is positively correlated to the improvements brought by HSSL. This positive correlation motivates us to propose an efficient search strategy that finds the most suitable auxiliary head for a specific model and several simple but effective designs to enlarge the model discrepancy. We show that HSSL is orthogonal to different self-supervised learning methods and boosts the performance on various downstream tasks, including image classification, semantic segmentation, object detection, and instance segmentation.

Limitations and further works. HSSL has been successfully integrated with various self-supervised learning methods to enhance performance. However, certain methods, such as those in [66], which utilize representations extracted from a frozen pre-trained model as learning targets, are less straightforward to adapt. Future work could focus on developing more general or specialized approaches to effectively incorporate heterogeneous representation learning with a broader range of methods. Additionally, HSSL measures model discrepancy using the Kullback-Leibler divergence between the probability distributions output by different models. While this metric is well-suited for clustering-based self-supervised learning, alternative metrics, such as CKA similarity [136], could be explored to evaluate discrepancy for other self-supervised learning methods that output representations. Regarding the searching strategy, future research can also aim to design more accurate and efficient search strategies that accommodate a wider range of models, including those with complex structures, further broadening the applicability and scalability.

References

[1] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *ECCV*, 2018.

[2] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *IEEE ICCV*, 2021.

[3] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *ECCV*, 2020.

[4] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," in *ICLR*, 2021.

[5] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *IEEE CVPR*, 2022.

[6] Z. Huang, X. Jin, C. Lu, Q. Hou, M.-M. Cheng, D. Fu, X. Shen, and J. Feng, "Contrastive masked autoencoders are stronger vision learners," *IEEE TPAMI*, vol. 46, no. 4, pp. 2506–2517, 2024.

[7] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *ICML*, 2022.

[8] Z.-Y. Li, S. Gao, and M.-M. Cheng, "Exploring feature self-relation for self-supervised transformer," *IEEE TPAMI*, 2022.

[9] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *NeurIPS*, 2020.

[10] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, "ibot: Image bert pre-training with online tokenizer," in *ICLR*, 2022.

[11] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Ávila Pires, Z. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - a new approach to self-supervised learning," in *NeurIPS*, 2020.

[12] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *IEEE ICCV*, 2021.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[14] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, pp. 303–338, 2009.

[15] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, P. Sun, Z. Li, and P. Luo, "Detco: Unsupervised contrastive learning for object detection," in *IEEE ICCV*, 2021.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[17] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *IEEE ICCV*, 2019.

[18] H. Yan, Z. Li, W. Li, C. Wang, M. Wu, and C. Zhang, "Contnet: Why not use convolution and transformer at the same time?" *arXiv preprint arXiv:2104.13497*, 2021.

[19] C. Ge, Y. Liang, Y. Song, J. Jiao, J. Wang, and P. Luo, "Revitalizing cnn attentions via transformers in self-supervised visual representation learning," in *NeurIPS*, 2021.

[20] B. Yin, X. Zhang, Z. Li, L. Liu, M.-M. Cheng, and Q. Hou, "Dformer: Rethinking rgbd representation learning for semantic segmentation," *arXiv preprint arXiv:2309.09668*, 2023.

[21] S. D'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," in *ICML*, 2021.

[22] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH*, 2020.

[23] X. Pan, C. Ge, R. Lu, S. Song, G. Chen, Z. Huang, and G. Huang, "On the integration of self-attention and convolution," in *IEEE CVPR*, 2022.

[24] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *IEEE ICCV*, 2021.

[25] S. Gao, Z.-Y. Li, M.-H. Yang, M.-M. Cheng, J. Han, and P. Torr, "Large-scale unsupervised semantic segmentation," *IEEE TPAMI*, 2022.

[26] B. Sun, Y. Yang, W. Yuan, L. Zhang, M.-M. Cheng, and Q. Hou, "Corrmatch: Label propagation via correlation matching for semi-supervised semantic segmentation," *arXiv preprint arXiv:2306.04300*, 2023.

[27] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *ECCV*, 2016.

[28] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *IEEE CVPR*, 2017.

[29] M. Noroozi and P. Favaro, "Unsupervised learning of visual representions by solving jigsaw puzzles," in *ECCV*, 2016.

[30] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *ICLR*, 2018.

[31] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *IEEE ICCV*, 2015.

[32] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*, 2008.

[33] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *IEEE CVPR*, 2016.

[34] M. Noroozi, H. Pirsiavash, and P. Favaro, "Representation learning by learning to count," in *IEEE ICCV*, 2017.

[35] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *IEEE CVPR*, 2018.

[36] Y. Zhao, G. Wang, C. Luo, W. Zeng, and Z.-J. Zha, "Self-supervised visual representations learning by contrastive mask prediction," in *IEEE ICCV*, 2021.

[37] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations," in *IEEE ICCV*, 2021.

[38] S. A. Koohpayegani, A. Tejankar, and H. Pirsiavash, "Mean shift for self-supervised learning," in *IEEE ICCV*, 2021.

[39] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," *NeurIPS*, 2014.

[40] H. Bao, L. Dong, S. Piao, and F. Wei, "BEit: BERT pre-training of image transformers," in *ICLR*, 2022.

[41] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: A simple framework for masked image modeling," in *IEEE CVPR*, June 2022, pp. 9653–9663.

[42] J. Liu, X. Huang, J. Zheng, Y. Liu, and H. Li, "Mixmae: Mixed and masked autoencoder for efficient pretraining of hierarchical vision transformers," in *IEEE CVPR*, 2023.

[43] J. Chen, M. Hu, B. Li, and M. Elhoseiny, "Efficient self-supervised vision pretraining with local masked reconstruction," *arXiv preprint arXiv:2206.00790*, 2022.

[44] W. Li, J. Xie, and C. C. Loy, "Correlational image modeling for self-supervised visual pre-training," in *IEEE CVPR*, 2023.

[45] Y. Fang, L. Dong, H. Bao, X. Wang, and F. Wei, "Corrupted image modeling for self-supervised visual pre-training," in *ICLR*, 2023.

[46] B. Roh, W. Shin, I. Kim, and S. Kim, "Spatilly consistent representation learning," in *IEEE CVPR*, 2021.

[47] O. J. Hénaff, S. Koppula, J.-B. Alayrac, A. van den Oord, O. Vinyals, and J. a. Carreira, "Efficient visual pretraining with contrastive detection," in *IEEE ICCV*, 2021.

[48] P. Zhou, Y. Zhou, C. Si, W. Yu, T. K. Ng, and S. Yan, "Mugs: A multi-granular self-supervised learning framework," in *arXiv preprint arXiv:2203.14415*, 2022.

[49] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, 2016.

[50] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *IEEE CVPR*, 2020.

[51] C.-H. Yeh, C.-Y. Hong, Y.-C. Hsu, T.-L. Liu, Y. Chen, and Y. LeCun, "Decoupled contrastive learning," in *ECCV*, 2022.

[52] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[53] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense contrastive learning for self-supervised visual pre-training," in *IEEE CVPR*, 2021.

[54] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, "Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning," in *IEEE CVPR*, 2021.

[55] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," in *ICML*, 2020.

[56] X. Chen and K. He, "Exploring simple siamese representation learning," in *IEEE CVPR*, 2021.

[57] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe, "Whitening for self-supervised representation learning," in *ICML*, 2021.

[58] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong, and C. C. Loy, "Online deep clustering for unsupervised representation learning," in *IEEE CVPR*, 2020.

[59] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *IEEE CVPR*, 2016.

[60] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," *arXiv:2304.07193*, 2023.

[61] K. Zhu, M. Fu, and J. Wu, "Multi-label self-supervised learning with scene images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[62] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin, "Unsupervised pre-training of image features on non-curated data," in *IEEE ICCV*, 2019.

[63] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *ICML*, 2021.

[64] A. Bardes, J. Ponce, and Y. LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," in *ICLR*, 2022.

[65] N. Shvetsova, F. Petersen, A. Kukleva, B. Schiele, and H. Kuehne, "Learning by sorting: Self-supervised learning with group ordering constraints," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 16 453–16 463.

[66] S. Gao, P. Zhou, M.-M. Cheng, and S. Yan, "Towards sustainable self-supervised learning," *arXiv preprint arXiv:2210.11016*, 2022.

[67] Y. M. Asano, C. Rupprecht, and A. Vedaldi, "Self-labelling via simultaneous clustering and representation learning," in *ICLR*, 2020.

[68] L. Wang, F. Liang, Y. Li, H. Zhang, W. Ouyang, and J. Shao, "Repre: Improving self-supervised vision transformer with reconstructive pre-training," in *IJCAI*, 7 2022, pp. 1437–1443.

[69] K. Song, S. Zhang, Z. Luo, T. Wang, and J. Xie, "Semantics-consistent feature search for self-supervised visual representation learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[70] W. Zhang, J. Pang, K. Chen, and C. C. Loy, "Dense siamese network for dense unsupervised learning," in *ECCV*, 2022.

[71] P. O. O Pinheiro, A. Almahairi, R. Benmalek, F. Golemo, and A. C. Courville, "Unsupervised learning of dense visual representations," *NeurIPS*, 2020.

[72] J. Xie, X. Zhan, Z. Liu, Y. S. Ong, and C. C. Loy, "Unsupervised object-level representation learning from scene images," in *NeurIPS*, 2021.

[73] L. Huang, S. You, M. Zheng, F. Wang, C. Qian, and T. Yamasaki, "Learning where to learn in cross-view self-supervised learning," in *CVPR*, 2022.

[74] I. Kakogeorgiou, S. Gidaris, B. Psomas, Y. Avrithis, A. Bursuc, K. Karantzalos, and N. Komodakis, "What to hide from your students: Attention-guided masked image modeling," in *ECCV*, 2022.

[75] Z. Li, Z. Chen, F. Yang, W. Li, Y. Zhu, C. Zhao, R. Deng, L. Wu, R. Zhao, M. Tang, and J. Wang, "MST: Masked self-supervised transformer for visual representation," in *NeurIPS*, 2021.

[76] C. Li, J. Yang, P. Zhang, M. Gao, B. Xiao, X. Dai, L. Yuan, and J. Gao, "Efficient self-supervised vision transformers for representation learning," in *ICLR*, 2022.

[77] H. Wang, K. Song, J. Fan, Y. Wang, J. Xie, and Z. Zhang, "Hard patches mining for masked image modeling," in *IEEE CVPR*, 2023.

[78] Z. Feng and S. Zhang, "Evolved part masking for self-supervised learning," in *IEEE CVPR*, 2023.

[79] G. Li, H. Zheng, D. Liu, C. Wang, B. Su, and C. Zheng, "Semmae: Semantic-guided masking for learning masked autoencoders," in *NeurIPS*, 2022.

[80] J. Xie, W. Li, X. Zhan, Z. Liu, Y. S. Ong, and C. C. Loy, "Masked frequency modeling for self-supervised visual pre-training," in *ICLR*, 2023.

[81] Y. Liu, S. Zhang, J. Chen, Z. Yu, K. Chen, and D. Lin, "Improving pixel-based mim by reducing wasted modeling capability," in *IEEE ICCV*, 2023.

[82] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE CVPR*, 2005.

[83] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *IEEE CVPR*, 2022.

[84] H. Wang, Y. Tang, Y. Wang, J. Guo, Z.-H. Deng, and K. Han, "Masked image modeling with local multi-scale reconstruction," in *IEEE CVPR*, 2023.

[85] H. Liu, X. Jiang, X. Li, A. Guo, Y. Hu, D. Jiang, and B. Ren, "The devil is in the frequency: Geminated gestalt autoencoder for self-supervised visual pre-training," in *AAAI*, 2023.

[86] H. Wang, J. Fan, Y. Wang, K. Song, T. Wang, and Z. Zhang, "Droppos: Pre-training vision transformers by reconstructing dropped positions," in *NeurIPS*, 2023.

[87] K. Yi, Y. Ge, X. Li, S. Yang, D. Li, J. Wu, Y. Shan, and X. Qie, "Masked image modeling with denoising contrast," *ICLR*, 2023.

[88] C. Tao, X. Zhu, W. Su, G. Huang, B. Li, J. Zhou, Y. Qiao, X. Wang, and J. Dai, "Siamese image modeling for self-supervised vision representation learning," in *IEEE CVPR*, 2023.

[89] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *IEEE CVPR*, 2023.

[90] Y. Chen, Y. Liu, D. Jiang, X. Zhang, W. Dai, H. Xiong, and Q. Tian, "Sdae: Self-distillated masked autoencoder," in *ECCV*, 2022.

[91] X. Chen, M. Ding, X. Wang, Y. Xin, S. Mo, Y. Wang, S. Han, P. Luo, G. Zeng, and J. Wang, "Context autoencoder for self-supervised representation learning," *IJCV*, vol. 132, no. 1, pp. 208–223, 2024.

[92] X. Dong, J. Bao, T. Zhang, D. Chen, W. Zhang, L. Yuan, D. Chen, F. Wen, N. Yu, and B. Guo, "Peco: Perceptual codebook for bert pre-training of vision transformers," in *AAAI*, 2023.

[93] P. Gao, Z. Lin, R. Zhang, R. Fang, H. Li, H. Li, and Y. Qiao, "Mimic before reconstruct: Enhancing masked autoencoders with feature mimicking," *IJCV*, vol. 132, no. 5, pp. 1546–1556, 2024.

[94] X. Dong, J. Bao, T. Zhang, D. Chen, W. Zhang, L. Yuan, D. Chen, F. Wen, and N. Yu, "Bootstrapped masked autoencoders for vision bert pretraining," in *ECCV*, 2022.

[95] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva: Exploring the limits of masked visual representation learning at scale," in *IEEE CVPR*, 2023.

[96] xingbin liu, J. Zhou, T. Kong, X. Lin, and R. Ji, "Exploring target representations for masked autoencoders," in *ICLR*, 2024.

[97] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas, "Masked siamese networks for label-efficient learning," in *ECCV*, 2022.

[98] Z. Wu, Z. Lai, X. Sun, and S. Lin, "Extreme masking for learning instance and distributed visual representations," *arXiv preprint arXiv:2206.04667*, 2022.

[99] Z. Jiang, Y. Chen, M. Liu, D. Chen, X. Dai, L. Yuan, Z. Liu, and Z. Wang, "Layer grafted pre-training: Bridging contrastive learning and masked image modeling for label-efficient representations," in *ICLR*, 2023.

[100] Y. Shi, N. Siddharth, P. H. Torr, and A. R. Kosiorek, "Adversarial masking for self-supervised learning," in *ICML*, 2022.

[101] S. Zhang, F. Zhu, R. Zhao, and J. Yan, "Contextual image masking modeling via synergized contrasting without view augmentation for faster and better visual pretraining," in *ICLR*, 2023.

[102] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *IEEE CVPR*, June 2023, pp. 16 133–16 142.

[103] L. Huang, S. You, M. Zheng, F. Wang, C. Qian, and T. Yamasaki, "Green hierarchical vision transformer for masked image modeling," in *NeurIPS*, 2022.

[104] P. Gao, T. Ma, H. Li, Z. Lin, J. Dai, and Y. Qiao, "MCMAE: Masked convolution meets masked autoencoders," in *NeurIPS*, 2022.

[105] X. Li, W. Wang, L. Yang, and J. Yang, "Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality," *arXiv:2205.10063*, 2022.

[106] K. Tian, Y. Jiang, Q. Diao, C. Lin, L. Wang, and Z. Yuan, "Designing bert for convolutional networks: Sparse and hierarchical masked modeling," in *ICLR*, 2023.

[107] Y. Tian, L. Xie, Z. Wang, L. Wei, X. Zhang, J. Jiao, Y. Wang, Q. Tian, and Q. Ye, "Integrally pre-trained transformer pyramid networks," in *IEEE CVPR*, 2023.

[108] Z. Peng, L. Dong, H. Bao, Q. Ye, and F. Wei, "A unified view of masked image modeling," *arXiv preprint arXiv:2210.10615*, 2022.

[109] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[110] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *IEEE CVPR*, 2022.

[111] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016.

[112] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, A. Joulin, G. Synnaeve, J. Verbeek, and H. Jégou, "Resmlp: Feedforward networks for image classification with data-efficient training," *IEEE TPAMI*, vol. 45, no. 4, pp. 5314–5321, 2023.

[113] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," in *IEEE CVPR*, 2022.

[114] C. Yang, Y. Wang, J. Zhang, H. Zhang, Z. Wei, Z. Lin, and A. Yuille, "Lite vision transformer with enhanced self-attention," in *IEEE CVPR*, 2022.

[115] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *IEEE CVPR*, June 2019.

[116] F. Yuan, Z. Zhang, and Z. Fang, "An effective cnn and transformer complementary network for medical image segmentation," *PR*, vol. 136, p. 109228, 2023.

[117] B. Yin, X. Zhang, Q. Hou, B.-Y. Sun, D.-P. Fan, and L. Van Gool, "Camoformer: Masked separable attention for camouflaged object detection," *arXiv preprint arXiv:2212.06570*, 2022.

[118] Y. Li, H. Mao, R. B. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *ECCV*, 2022, pp. 280–296.

[119] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," in *ICLR*, 2022.

[120] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *IEEE ICCV*, 2021.

[121] S. Lao, Y. Gong, S. Shi, S. Yang, T. Wu, J. Wang, W. Xia, and Y. Yang, "Attentions help cnns see better: Attention-based hybrid image quality assessment network," in *CVPRW*, 2022.

[122] K. Song, J. Xie, S. Zhang, and Z. Luo, "Multi-mode online knowledge distillation for self-supervised visual representation learning," in *IEEE CVPR*, 2023.

[123] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE CVPR*, 2020.

[124] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[125] Y. Li, Z.-Y. Li, Q.-S. Zeng, Q. Hou, and M.-M. Cheng, "Cascade-clip: Cascaded vision-language embeddings alignment for zero-shot semantic segmentation," in *ICML*, 2024.

[126] Z. Xie, Z. Geng, J. Hu, Z. Zhang, H. Hu, and Y. Cao, "Revealing the dark secrets of masked image modeling," in *IEEE CVPR*, 2023.

[127] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *ECCV*, September 2018.

[128] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *IEEE CVPR*, 2017.

[129] K. Chen, Z. Liu, L. Hong, H. Xu, Z. Li, and D.-Y. Yeung, "Mixed autoencoder for self-supervised visual representation learning," in *IEEE CVPR*, 2023.

[130] Z.-Y. Li, X. Jin, B. Sun, C.-L. Guo, and M.-M. Cheng, "Towards raw object detection in diverse conditions," *arXiv preprint arXiv:2411.15678*, 2024.

[131] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep. 0, 2009.

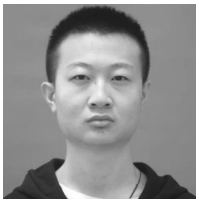
[132] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The inaturalist species classification and detection dataset," in *IEEE CVPR*, June 2018.

[133] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *IEEE ICCV*, 2021, pp. 568–578.

[134] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *IEEE CVPR*, June 2018.

[135] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" in *NeurIPS*, 2021.

[136] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *ICML*, 2019.



Zhong-Yu Li is a Ph.D. student from the college of computer science, Nankai university. He is supervised via Prof. Ming-Ming cheng. His research interests include deep learning, representation learning and computer vision.

Bo-Wen Yin is a Ph.D. student from the college of computer science, Nankai university. He is supervised via Prof. Qibin Hou. His research interests include computer vision and multimodal scene perception.

Yongxiang Liu received his Ph.D. degree in Information and Communication Engineering from the National University of Defense Technology (NUDT) in 2004. Currently, he is a full professor in the College of Electronic Science and Technology, NUDT. His research interests mainly include Remote Sensing Imagery Analysis, Radar Signal Processing, Synthetic Aperture Radar (SAR) Object Recognition, Inverse SAR (ISAR) Imaging, and Machine Learning.

Li Liu (Senior Member, IEEE) received her Ph.D. degree from the National University of Defense Technology (NUDT) in 2012. She is currently a full professor with the College of Electronic Science and Technology, NUDT. Her articles have currently over 11, 000 citations in Google Scholar. Her research interests include Computer Vision, Machine Learning, Trustworthy Artificial Intelligence, and Synthetic Aperture Radar. Dr. Liu served as the Guest Editor for special issues in IEEE TPAMI and the IJCV, and serves as an Associate Editor for IEEE TGRS, IEEE TCSVT, etc.

Ming-Ming Cheng (Senior Member, IEEE) received his PhD degree from Tsinghua University in 2012, and then worked with Prof. Philip Torr in Oxford for 2 years. Since 2016, he is a full professor at Nankai University, leading the Media Computing Lab. His research interests include computer vision and computer graphics. He received awards, including ACM China Rising Star Award, IBM Global SUR Award, *etc*. He is a senior member of the IEEE and on the editorial boards of IEEE TPAMI and IEEE TIP.