

Exploring Transferability of Self-Supervised Learning by Task Conflict Calibration

Huijie Guo^{1*}, Jingyao Wang^{1,2*}, Peizheng Guo^{1,2}, Xingchen Shen¹, Changwen Zheng^{1,2†}, Wenwen Qiang^{1,2,‡}

¹Institute of Software Chinese Academy of Sciences,

²University of Chinese Academy of Sciences,

{guohuijie,wangjingyao2023,guopeizheng2025,xingchen,changwen,qiangwenwen}@iscas.ac.cn

Abstract

In this paper, we explore the transferability of SSL by addressing two central questions: (i) what is the representation transferability of SSL, and (ii) how can we effectively model this transferability? Transferability is defined as the ability of a representation learned from one task to support the objective of another. Inspired by the meta-learning paradigm, we construct multiple SSL tasks within each training batch to support explicitly modeling transferability. Based on empirical evidence and causal analysis, we find that although introducing task-level information improves transferability, it is still hindered by task conflict. To address this issue, we propose a Task Conflict Calibration (TC²) method to alleviate the impact of task conflict. Specifically, it first splits batches to create multiple SSL tasks, infusing task-level information. Next, it uses a factor extraction network to produce causal generative factors for all tasks and a weight extraction network to assign dedicated weights to each sample, employing data reconstruction, orthogonality, and sparsity to ensure effectiveness. Finally, TC² calibrates sample representations during SSL training and integrates into the pipeline via a two-stage bi-level optimization framework to boost the transferability of learned representations. Experimental results on multiple downstream tasks demonstrate that our method consistently improves the transferability of SSL models.

Code — <https://github.com/PaulGHJ/TC2>

Introduction

Traditional unsupervised learning lacks ground-truth labels and often performs worse than supervised learning. However, supervised methods heavily rely on costly manual annotations and still face generalization issues (Bousquet and Elisseeff 2002; Mignacco et al. 2020). As a compelling alternative, Self-Supervised Learning (SSL) has shown strong performance in image classification and object recognition (Schiappa, Rawat, and Shah 2023; Sun et al. 2025; Liu et al. 2022b), sometimes surpassing supervised methods.

*These authors contributed equally.

†Corresponding Author.

‡CO-Corresponding Author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Existing self-supervised learning (SSL) methods primarily focus on designing reliable supervisory signals or introducing inductive biases to guide the model in learning effective feature representations. For example, discriminative SSL (D-SSL) methods construct positive pairs by generating different augmented views of the same sample and treat other samples as negatives (Chen et al. 2020; He et al. 2020; Caron et al. 2020). Generative SSL (G-SSL) methods mask part of the input and train the model to reconstruct the masked regions, using the reconstruction error as the supervisory signal (He et al. 2022; Bao et al. 2021). However, these methods often overlook a critical question: how do such learning paradigms ensure the transferability of learned features? This question is not easy to answer and helps explain why these methods often perform poorly on out-of-distribution (OOD) data or transfer tasks. Additionally, this limits the theoretical insights and methodological advances needed to improve their transfer generalization capabilities.

In this paper, we focus on two key questions: (1) What is representation transferability? (2) How can it be effectively modeled? For the first question, we take an intuitive perspective: if a representation has good transferability, it should perform well across various downstream tasks. Based on this intuition, we define task-level transferability as the ability of a representation learned from one task to support the objective of another (Yosinski et al. 2014). To address the second question, we consider meta-learning as an established approach to modeling transferability. Prior works (Ni et al. 2021) suggest that SSL can be regarded as a special case of meta-learning. The main difference lies in task organization within each mini-batch and learning mechanism. Meta-learning typically involves multiple tasks, whereas SSL is generally considered to involve only a single task. Moreover, meta-learning is optimized in a bi-level manner. It achieves strong transferability through multi-task bi-level training, which implicitly models a distribution over tasks. This allows meta-learning to generalize well to unseen tasks. Motivated by this, we propose enhancing the transferability of SSL by introducing a meta-learning mechanism. Specifically, we construct multiple tasks within each mini-batch during training and train SSL models in a bi-level manner.

We further explore the effectiveness of the above meta-learning-like SSL method on transferability. In the experi-

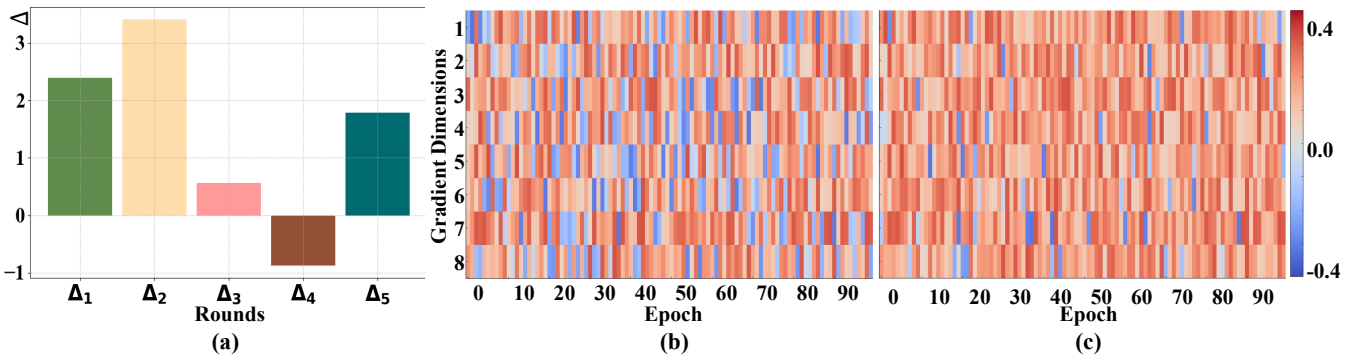


Figure 1: Effect of task-level information on CIFAR-100. (a) shows the effect of SimCLR trained on ImageNet and tested on CIFAR-100 before and after introducing task-level information in five runs. (b) and (c) show the training process corresponding to the worst and best rounds in five evaluations, visualizing the gradient similarity between tasks.

ment, we uniformly split each training batch into two subsets, which serve as two independent SSL tasks. Following the standard SSL training framework, each subset generates two augmented views via random data augmentation for training. We use SimCLR as the baseline, training on ImageNet-100 and testing on other datasets. First, we evaluate the Top-1 accuracy without task construction. Next, we assess performance with the multi-task learning mechanism through five independent training rounds and calculate the improvement for each round. From Figure 1(a), the results demonstrate that incorporating task-level information significantly improves model performance, but the improvement is not consistent, with a nearly 3% performance gap between the best and worst outcomes, and some cases even showing negative gains. Further analysis reveals that gradient conflict between tasks during training significantly affects transferability, as shown in Figure 1(b) and 1(c). Therefore, when introducing task-level information enhances transferability, inter-task gradient conflict remains an important issue.

To better understand the underlying causes of task conflict and its impact on transferability, we analyze it through both data generation and causal inference. Specifically, we construct a Structural Causal Model (SCM) to illustrate these relationships (as shown in Figure 2a), where Y_i, Y_j are task labels, $F_{i,j}^s$ is the shared generative factors, and F_i^u, F_j^u are task-specific factors. In meta-learning-like SSL, the model estimates generative factors from multiple tasks and predicts labels accordingly. Due to the shared training across tasks, the extracted factors inevitably mix semantics from all tasks (Figure 2b). As shown in Theorem 1, if task-specific factors are not correctly identified, these factors from other tasks act as confounders, hindering target task learning—this phenomenon is termed task conflict. It degrades per-task learning and limits the effectiveness of modeling task distributions, explaining the inconsistent transfer performance observed in meta-learning-like SSL. Moreover, these generative factors can be seen as semantic vectors composing the feature space. Since different tasks may involve distinct semantics, task conflict implies that target features may be contaminated by irrelevant semantics. Thus, enhancing SSL transferability requires calibrating each sam-

ple’s feature representation to retain only task-relevant semantics within the meta-learning-like framework.

Based on the above analysis, we propose a new method, called Task Conflict Calibration (TC^2), to alleviate task conflict when explicitly modeling the transferability for SSL methods. First, TC^2 constructs multiple SSL tasks through batch splitting, which introduces task-level information into the model. Second, TC^2 learns a factor extraction network f_v to generate causal generative factors for all tasks and a weight extraction network f_w to generate a weight of a single sample of each task. We leverage techniques such as data reconstruction, orthogonality, and sparsity to ensure that the learned f_v can effectively generate causal generative factors suitable for multiple tasks, while the learned function f_w produces a weight matrix of semantic vectors that are specifically relevant to the target task. Third, based on f_v and f_w , we formulate the final version of TC^2 , which serves to calibrate sample representations during the SSL training process. Finally, we embed TC^2 into the SSL training pipeline through a two-stage bi-level optimization framework to enhance the transferability of learned representations. Extensive transfer learning and ablation experiments demonstrate the effectiveness of TC^2 . Our contributions include:

- We find that simply formulating the SSL training process as a meta-learning paradigm does not guarantee improved feature transferability. Through causal analysis, we demonstrate this limitation stems from task conflict.
- We propose a method to alleviate task conflict and improve the transferability of SSL models, leveraging a two-stage bi-level optimization mechanism.
- Experimental results on multiple downstream tasks demonstrate the effectiveness of our method while preserving the model’s inherent representation capabilities.

Related Work

Self-Supervised Learning (SSL) has demonstrated remarkable performance in multiple fields, even rivaling supervised learning. Early SSL methods focused on designing pretexts (Albelwi 2022; Doersch, Gupta, and Efros 2015) to provide supervisory information for the unlabeled data.

With the rise of contrastive learning, SSL gradually turns to learning more generalizable representations based on invariance to data augmentation. Typical contrastive learning constructs positive sample pairs through data augmentation so that positive pairs are close and negative pairs are separated in the embedding space. SimCLR (Chen et al. 2020) introduces a simple yet effective contrastive learning framework, where the model is trained to maximize the consistency between positive pairs and minimize the consistency between negative pairs through the InfoNCE loss (Gutmann and Hyvärinen 2010). To reduce the computational cost caused by batch size, MoCo (He et al. 2020) uses a memory bank to increase the number of negative samples and a momentum encoder to learn representations. SwAV (Caron et al. 2020) enforces consistency between cluster assignments from different augmented views of the same image, rather than computing comparisons of sample pairs. BYOL (Grill et al. 2020) and SimSiam (Chen and He 2021) only constrain the consistency between positive samples to avoid trivial solutions through the online and target networks. Additionally, some studies aim to reduce redundancy among features (Liu et al. 2022a; Bardes, Ponce, and Lecun 2022). Barlow Twins (Zbontar et al. 2021) endeavors to make the normalized cross-correlation matrix of the augmented embeddings close to the identity matrix.

Transferability refers to the ability of a model trained on one task to be transferred to another (Pan and Yang 2009; Sun et al. 2024b; Zang et al. 2025). The interplay between out-of-distribution and transfer performance of convolutional neural networks is investigated by (Djolonga et al. 2020). According to (Ying et al. 2018), achieving optimal transferability often requires exhaustive search or the use of diverse task distributions. Multi-task learning (Standley et al. 2020) makes the model more adaptable by sharing knowledge among multiple related tasks, thereby improving its transferability to new ones. The connection between the structure of vision tasks and transferability is highlighted by (Zamir et al. 2018), which explores which tasks can be effectively transferred to any target task. To enable fast adaptation to new tasks with limited labels, meta-learning leverages many small-sample tasks to train a base learner (Hospedales et al. 2021; Sun et al. 2021). The connection between SSL and meta-learning to improve the transferability of self-supervised models is explored by some work (Ni et al. 2021).

Existing SSL methods construct models based on samples, which can be treated as an in-distribution problem for a specific task, without explicitly modeling transferability from the task level. In this paper, we explicitly model transferability for SSL by constructing multiple tasks and improving the transferability by resolving task conflict issues.

Problem Analysis and Motivation

This section begins by revisiting the SSL paradigm, highlighting the implicit task structure underlying it. We then explore the inherent connection between SSL and meta-learning through the lens of task modeling. Building on this, we empirically investigate how introducing task-level information can lead to task conflicts, which hinder the transferability of learned representations. Finally, we analyze the

root causes of these conflicts from a causal perspective.

Rethinking SSL from a Task Perspective

In the training phase, data is organized into mini-batches $\mathcal{X}_{tr} = \{x_i\}_{i=1}^N$. In D-SSL methods such as SimCLR (Chen et al. 2020) and Barlow Twins (Zbontar et al. 2021), each sample is stochastically augmented into two views (x_i^1, x_i^2) . In G-SSL methods like MAE (He et al. 2022), x_i is divided into patches, masked, and reassembled into x_i^1 , while the original sample serves as x_i^2 . Thus, the augmented training set becomes $\mathcal{X}_{tr}^{aug} = \{(x_i^1, x_i^2)\}_{i=1}^N$, and SSL aims to learn a feature extractor f from these pairs.

D-SSL objectives typically consist of alignment (maximizing similarity within pairs) and regularization (e.g., enforcing uniformity (Wang and Isola 2020)). G-SSL achieves alignment by reconstructing x_i^2 from x_i^1 using an encoder-decoder structure. In both cases, one sample in the pair serves as an *anchor*, guiding the other to align with it. For D-SSL, this is explicit; for G-SSL, x_i^2 acts as the anchor for reconstructing x_i^1 . This process can be interpreted as assigning one sample as a learning target and adjusting the other to match it in feature space. As a result, paired samples become tightly clustered, with the anchor acting like a cluster center. Consequently, \mathcal{X}_{tr}^{aug} can be viewed as a mini-batch classification task with N implicit classes, where alignment functions similarly to class-specific feature learning.

Comparison of SSL and Meta-Learning

During the meta-training phase of meta-learning, the model is trained on a collection of tasks sampled from a task distribution. Each training mini-batch consists of multiple tasks $\{\tau_1, \tau_2, \dots, \tau_B\}$, where each task τ_i contains a support set S_i and a query set Q_i . The support set is used to adapt the model to the specific task, while the query set is used to evaluate and update the meta-learner’s parameters based on the adaptation performance. The learning mechanism of meta-learning typically follows a bi-level optimization paradigm (Gordon et al. 2018; Jiang et al. 2022): the inner loop adapts task-specific parameters using the support set, and the outer loop updates the shared meta-parameters using the query set loss across all tasks in the mini-batch. This enables the model to acquire generalizable knowledge that can be quickly adapted to unseen tasks during meta-testing.

Comparing SSL and meta-learning, we can obtain SSL and meta-learning share certain similarities in task construction but differ significantly in mini-batch construction and learning mechanism (Ni et al. 2021). Based on the above analysis, we can obtain the collection of all instance pairs across the entire training set can be viewed as forming a multi-class classification task. Similar to traditional machine learning, the training process of SSL can be viewed as a mini-batch-based learning procedure. The basic unit of its training set is a sample pair, and multiple distinct pairs form a mini-batch. The training process essentially aims to approximate the conditional distribution $p(y|x)$. Notably, the learned distribution $p(y|x_{train})$ can generalize to $p(y|x_{test})$ only under the assumption that training and test data are i.i.d.. However, in OOD or transfer tasks, this i.i.d. assumption often fails, which poses a challenge to the performance

of SSL in such scenarios. In contrast, meta-learning adopts a different structural unit: each basic unit in the training set corresponds to an entire task, and a mini-batch consists of multiple such tasks. During meta-training, the model learns to approximate the same $p(y|x)$ across all samples within any given task. Since both training and test tasks are assumed to be sampled from the same (possibly unknown) task distribution, the i.i.d. assumption naturally holds at the task level. As a result, the function learned during meta-training can be effectively transferred to test tasks, making meta-learning more robust and reliable when facing OOD problem or transfer learning scenarios.

Empirical Evidence

One promising way to improve the transferability of SSL representations is to incorporate task-level information inspired by meta-learning. Specifically, we uniformly split each training batch \mathcal{X}_{tr}^{aug} into two subsets B_1 and B_2 , which serve as two distinct tasks. Each task is further divided into a support set B_i^s and query set B_i^q , where $i \in \{1, 2\}$, following the meta-learning framework. We adopt SimCLR as the baseline, training on ImageNet-100 and testing on multiple datasets. Here we just report results on CIFAR-100 (Details are provided in the Appendix). First, we evaluate the Top-1 accuracy without task construction, denoted as $acc(\text{SimCLR})$. Then, using the task-based multi-task learning mechanism, we conduct five independent runs, with the i -th result denoted as $acc(\text{SimCLR} + T, i)$. The corresponding performance gain is $\Delta_i = acc(\text{SimCLR} + T, i) - acc(\text{SimCLR})$. Results in Figure 1(a) show that task-level design improves performance, but with noticeable instability—performance differences can reach nearly 3%, and negative gains are observed in some runs.

To investigate this instability, we analyze the first 100 training epochs of the best and worst performing runs. We compute the cosine similarity of gradients between tasks at each epoch. A negative value indicates conflicting gradients (angle $> 90^\circ$), while a positive value indicates alignment. As shown in Figure 1(b) and Figure 1(c), we observe that: (i) gradient similarity fluctuates during training, often exhibiting conflict; (ii) lower inter-task gradient conflict correlates with better transfer performance. These results suggest that while task-level information enhances SSL transferability, it introduces inter-task gradient conflict, which can negatively affect consistency and stability in performance.

Causal Analysis and Motivation

Causal Analysis. To investigate the underlying reasons for the above phenomenon, we propose using causal theory for analysis. We construct a Structural Causal Model (SCM) based on the causal generating mechanism (Suter et al. 2019; Hu et al. 2022), as shown in Figure 2(a). Specifically, the SCM comprises two tasks, τ_i and τ_j , where Y_i and Y_j denote the label variables for τ_i and τ_j , and X_i and X_j represent the generated samples corresponding to these tasks. Meanwhile, F_i^u and F_j^u denote the sets of unique causal factors specific to τ_i and τ_j , respectively, and $F_{i,j}^s$ contains the shared causal generative factors. Following (Sun et al. 2024a; Deshpande et al. 2022), the samples are generated with all the causal

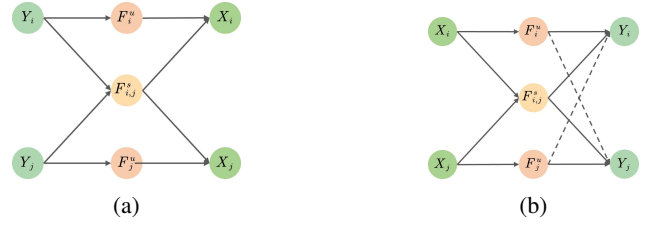


Figure 2: (a) Overview of the SCM based on generation mechanisms; (b) The causal mechanism of task conflict.

generative factors, e.g., for τ_i , the sample X_i is generated by both the task-specific factors F_i^u and the shared factors $F_{i,j}^s$. Since each causal generative factor in F_i^u , F_j^u , and $F_{i,j}^s$ corresponds to a specific semantic attribute (e.g., color, shape) in its respective task, they can be viewed as being determined by the label variable Y_i (or Y_j) of the tasks. Thus, we obtain Figure 2a, and for task τ_i , we refer to $F_{i,j}^s$ and F_i^u as the causal generating factors that are causally related to Y_i , while F_j^u is considered non-causal factors.

Based on the above SCM, a good SSL model should utilize only the causal factors to learn the tasks, thereby achieving accurate decisions even in the absence of labels. However, to achieve modeling transferability, a multi-task joint learning mechanism is required, which causes the model to learn all the causal factors across tasks, including F_i^u , F_j^u , and $F_{i,j}^s$. Consequently, for a specific task τ_i , the model may learn from non-causal factors originating from other tasks, which poses a significant challenge to attaining optimal predictions. To validate this assertion, we consider a scenario involving two tasks in the same SSL training batch. Here, we treat the pseudo-labels generated by data augmentation as Y_i and Y_j . Then, we have:

Theorem 1 *If the correlation between Y_i and Y_j is not equal to 0.5, then the optimal model for task i has a non-zero weight on F_j^u . If the correlation is equal to 0.5 with limited training samples, then the optimal classifier for task i also has non-zero weight on factor F_j^u .*

It shows that the learned SSL model leverages causal factors from other tasks to facilitate the learning of the target task (with proofs in Appendix). For example, in task τ_i , the model employs causal factors F_j^u from task τ_j to learn Y_i , creating a spurious correlation represented as $F_j^u \rightarrow Y_i$; similarly, $F_i^u \rightarrow Y_j$ exists for task τ_j . Therefore, the SCM for the learning phase, shown in Figure 2b, which contains two spurious paths. These spurious correlations lead to sub-optimal learning for each task. For any target task, interference from other tasks during training disrupts learning. A direct indicator of this is when the gradients between tasks have angles greater than 90° (Figure 1), meaning that updates from other tasks suppress the gradient updates of the target task—a phenomenon we refer to as “task conflict”.

Motivation. Based on the above causal analysis, enhancing SSL feature transferability requires not only constraining the learning mechanism as in meta-learning but also addressing task conflict during training. Generative factors can be viewed as semantic vectors (Pearl 2009; Wolff and

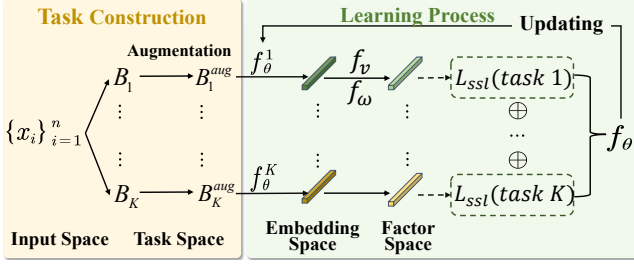


Figure 3: Overview of the proposed framework.

Zettergren 2019; Schölkopf et al. 2021), and sample features as projections onto them. Task conflict arises when the multi-task learning process forces the model to learn all task-related semantic vectors. As a result, a target task’s feature representation may contain non-zero projections onto irrelevant semantics from other tasks, degrading its effectiveness. To resolve this, we propose identifying the semantic vectors relevant to the target task and retaining only the corresponding feature projections. This approach effectively eliminates the interference of unrelated semantics, thereby mitigating task conflict and improving feature transferability.

Methodology

Task Construction

In this subsection, we describe how to construct SSL tasks, as shown in Figure 3. We begin by randomly sampling a mini-batch $X_{tr} = \{x_i\}_{i=1}^N$ from the training set.

Drawing inspiration from meta-learning (Vilalta and Drissi 2002; Finn, Abbeel, and Levine 2017), we divide the mini-batch X_{tr} into K groups, denoted as B_1, \dots, B_K , with each group treated as an individual task. This grouping strategy allows the model to simulate diverse tasks within a single batch, promoting the learning of task-agnostic representations. Such representations are crucial for improving the model’s transferability to unseen domains or tasks. For D-SSL, we randomly apply m data augmentation strategies to each group to generate the augmented dataset B_k^{aug} , containing m augmented views for each sample. Each task consists of a support set and a query set. The support set is constructed by randomly selecting m' augmented views from each sample, while the remaining views are assigned to the query set. The both sets are then used to update the task-specific model parameters and guide the learning process for each task. For G-SSL, each sample is first divided into multiple patches. Then, m different random masks are applied to produce m masked views. Each task consists of a support set and a query set. The support set is constructed by randomly selecting m' masked views for each sample, along with the unmasked sample. The remaining masked views are used to construct the query set. The model is trained to reconstruct the original input from the masked views by minimizing the reconstruction loss. By constructing multiple tasks, the SSL model can learn both the sample- and task-level information, improving its transferability. However, the inevitability of task conflicts may affect the transferability of the model, and we will address this issue in the next subsection.

Task Conflict Calibration

In this subsection, we propose a method for calibrating task conflict. Specifically, we address the following three key questions: (i) how to learn the union of causal generative factors across all tasks; (ii) how to identify the task-specific causal factors for individual tasks; and (iii) how to alleviate the conflicts that may arise between tasks.

For question (i), we first compute the mean vector and covariance matrix of all sample representations in the training batch. The mean vector represents the center of the sample features, while the covariance matrix captures the relationships between the features. We then concatenate these two statistical quantities to form a new matrix $S \in R^{(d+1) \times d}$, for the subsequent factor extraction process. Next, we fed the concatenated matrix into a factor extraction network f_v . The output of f_v is constrained to be a factor matrix $V = f_v(S) \in R^{d_v \times n_f}$, where each column represents the basis of a factor, d_v is the dimension of the generative factor (which is the same as the dimension of the sample representation, denoted as d), and n_f is the number of factors. At last, the loss function of learning f_v can be presented as:

$$\mathcal{L}_v(f_v) = \|V^T V - I\|_F^2 + \|Z_{tr}^{aug} - Z_{tr}^{aug} V V^T\|_F^2. \quad (1)$$

where Z_{tr}^{aug} represent the augmented dataset on the whole training batch, I is a identity matrix, $\|\cdot\|_F$ is the F -paradigm of the matrix. The first term is to constrain the orthogonality of the column vectors of V , and the second term is to constrain V to best reconstruct the training dataset.

Since the SSL task in this work is constructed by partitioning the entire training batch, and the full set encompasses all tasks, it is reasonable to base the learning on the entire training batch. This approach allows the model to capture the global characteristics of the data, facilitating generalization and transfer across tasks. Specifically, the mean vector and covariance matrix serve as sufficient statistics, summarizing the main features of the data and reflecting its overall distribution. By using these statistics to extract factors, the model learns shared representations that can adapt to different tasks, thereby improving transferability. As different factors represent distinct semantics, enforcing orthogonality among their corresponding vectors serves as a feasible approach to learning f_v . Minimizing the reconstruction error ensures that the factor matrix accurately reconstructs the data, allowing the shared factors across tasks to be effective and adaptable, further supporting transfer learning.

For question (ii), given a sample of a task, we input it into a learnable weight extraction network f_w . We constrain that the output of f_w is a weight vector with n_f dimension. Then, the loss function of learning f_w is presented:

$$\mathcal{L}_w(f_w) = \sum_k \sum_{z_j \in \text{task } k} \|z_j - W_j V^T\|_2^2 + \|W_j\|_1. \quad (2)$$

Here, V is a pre-given matrix, $W_j = f_w(z_j) V^T$ represents the weight vector of the j -th sample under task k , z_j is the representation of a sample. The first term ensures that each sample representation z_j is reconstructed by a weighted combination of a few shared factors in V , with W_j serving as its sample-specific weight vector. This encourages the

model to capture task-specific structures through selective factor usage. The second term applies an l_1 -norm regularization to W_j , promoting sparsity by activating only a limited number of factors. This improves interpretability and reduces overfitting. Together, these terms guide f_w to assign sparse, task-aware weights over shared factors, enabling the identification of task-specific causal components.

The essence of task conflict lies in the causal generative factors of the target task being confounded by factors from other tasks. A feasible solution is to constrain the features of samples from the target task to contain only semantics that are relevant to the target task itself. From the perspective of feature representation, this can be interpreted as enforcing that the features of a sample are generated solely by the causal generative factors associated with the target task. Therefore, based on the solutions to question (i) and (ii), the solution to question (iii) for any sample z_j in the target task can be directly formulated as: $f_w(z_j)V^T$. Finally, we get the task conflict calibration loss by combining Eq.1 and Eq.2:

$$\mathcal{L}_{tc^2}(f_v, f_w) = \mathcal{L}_v + \lambda_s \mathcal{L}_w, \quad (3)$$

where λ_s is the parameter used for balancing two terms.

Learning Process

The training process with TC² in each batch is performed via two stages. In this first stage, we fix the SSL model and optimize the transfer function f_v and weight function f_w via bi-level optimization. In the second stage, we optimize the SSL model with f_v and f_w being held.

In the first stage, we optimize f_v and f_w to ensure the causality of the learned generative factors. Following (Zhu, An, and Huang 2021; Deng, Gould, and Zheng 2022), generative factors should remain invariant across different sample distributions within a task. An effective model captures these factors and adapts across tasks. Support and query sets from the same task represent shifted distributions, and the meta-learning objective is to learn invariants effective across both.

Thus, we adopt bi-level optimization to learn f_v and f_w , enforcing causal invariance. First, we first update f_v and f_w on the support sets of all tasks by the following formula:

$$\begin{aligned} f'_v &\leftarrow f_v - \alpha_1 \nabla_{f_v} \bar{\mathcal{L}}^s, & f'_w &\leftarrow f_w - \alpha_2 \nabla_{f_w} \bar{\mathcal{L}}^s \\ s.t. \bar{\mathcal{L}}^s &= \frac{1}{K} \sum_{k=1}^K \mathcal{L}_{ssl}^s(f_w, f_v, B_k^{aug}) + \mathcal{L}_{tc^2}(f_v, f_w) \end{aligned} \quad (4)$$

Here, $\mathcal{L}_{ssl}^s(\cdot)$ is the self-supervised loss on support set S_k of task k , and α_1, α_2 are learning rates. Note that losses are computed in the factor space.

Next, f_v and f_w are updated on query sets:

$$\begin{aligned} f_v &\leftarrow f'_v - \alpha_3 \nabla_{f'_v} \bar{\mathcal{L}}^q, & f_w &\leftarrow f'_w - \alpha_4 \nabla_{f'_w} \bar{\mathcal{L}}^q \\ s.t. \bar{\mathcal{L}}^q &= \frac{1}{K} \sum_{k=1}^K \mathcal{L}_{sse}^q(f'_w, f'_v, B_k^{aug}) + \mathcal{L}_{tc^2}(f'_v, f'_w) \end{aligned} \quad (5)$$

$\mathcal{L}_{sse}^q(\cdot)$ denotes the sum of squared errors on query set Q_k , with learning rates α_3, α_4 .

In the second stage, with f_v and f_w fixed, we update f_θ via bi-level optimization to explicitly encode transferability. We first fine-tune f_θ for each task using its support set:

$$f_\theta^k \leftarrow f_\theta - \beta_1 \nabla_{f_\theta} \mathcal{L}_{ssl}^s(B_k^{aug}, f_\theta), \quad (6)$$

where β_1 is the learning rate. Then, we update f_θ by minimizing the total query loss across tasks:

$$f_\theta \leftarrow f_\theta - \beta_2 \nabla_{f_\theta} \sum_{k=1}^K \mathcal{L}_{sse}^q(B_k^{aug}, f_\theta^k), \quad (7)$$

with β_2 as the learning rate.

This two-stage bi-level optimization mitigates task conflicts that hinder learning, and enables multi-task modeling in SSL to enhance representation transferability.

Experiments

In this section, we conduct extensive experiments on benchmark datasets to evaluate the effectiveness of the proposed TC². More details and results are provided in the Appendix.

Transfer Learning

In this section, we study the transferability of our proposed method under different transfer tasks, covering 7 benchmark datasets. Note that we also validate the effectiveness of our proposed method for unsupervised learning, semi-supervised learning and generative tasks in the Appendix.

Video-based Task. We evaluate TC² on video tasks by transferring the pre-trained model to the UniTrack benchmark (Wang et al. 2021). Table 1 reports the results on five tasks using features from [layer3/layer4] of ResNet-50. The results show that our proposed method has improvements compared with existing self-supervised methods. SimCLR increase about 3% in the VOS (Perazzi et al. 2016), and BYOL increase over 2% in the MOT (Milan et al. 2016).

Object Detection and Instance Segmentation. We evaluate our method on VOC 07 (Everingham et al. 2010) and COCO (Lin et al. 2014) with the most commonly used protocol (Zbontar et al. 2021). Several different self-supervised methods are used for comparison. We report the comparison results before and after introducing our proposed method in Table 2, showing that TC² brings stable performance improvements on different transfer tasks. For example, BYOL+TC² achieves SOTA on the VOC 07 detection task, and VICReg+TC² achieves SOTA on the COCO instance segmentation task.

OOD Tasks. We also assess our method on benchmark datasets that address the OOD challenge, i.e., PACS (Xu, Xiao, and López 2019), ColoredMNIST (Nam et al. 2020), and OfficeHome (Venkateswara et al. 2017). Specifically, we assess the performance of SSL baselines both before and after incorporating the TC² across three datasets. For PACS, we train the SSL models on these domains (Photo, Sketch, and Cartoon) and test them on all four domains, with the average performance also reported. For ColoredMNIST, we adopt the experimental setup from (Gat et al. 2020), evaluating the ability to generalize to new classes after training on base classes. Lastly, for OfficeHome, we randomly select one domain as the source for training and another as the target domain. Labels for the source domain are predefined, whereas those for the target domain remain unknown. We then compare the performance of SimCLR with and without the introduction of TC². As shown in Table 3, Table 4 and Table 7, TC² significantly enhances performance.

Method	SOT		VOS		MOT		MOTS		PoseTrack
	AUC _{XCorr}	AUC _{DCF}	J-mean	IDF1	HOTA	IDF1	HOTA	IDF1	
SimCLR	47.3 / 51.9	61.3 / 50.7	60.5 / 56.5	66.9 / 75.6	57.7 / 63.2	65.8 / 67.6	67.7 / 69.5	72.3 / 73.5	
MoCo	50.9 / 47.9	62.2 / 53.7	61.5 / 57.9	69.2 / 74.1	59.4 / 61.9	70.6 / 69.3	71.6 / 70.9	72.8 / 73.9	
SwAV	49.2 / 52.4	61.5 / 59.4	59.4 / 57.0	65.6 / 74.4	56.9 / 62.3	68.8 / 67.0	69.9 / 69.5	72.7 / 73.6	
BYOL	48.3 / 55.5	58.9 / 56.8	58.8 / 54.3	65.3 / 74.9	56.8 / 62.9	70.1 / 66.8	70.8 / 69.3	72.4 / 73.8	
Barlow Twins	44.5 / 55.5	60.5 / 60.1	61.7 / 57.8	63.7 / 74.5	55.4 / 62.4	68.7 / 67.4	69.5 / 69.8	72.3 / 74.3	
SimCLR+TC ²	51.3 / 54.2	63.8 / 54.2	63.4 / 59.3	70.8 / 78.3	62.9 / 64.5	67.4 / 70.5	68.9 / 71.1	73.2 / 74.3	
BYOL+TC ²	51.6 / 58.1	59.9 / 58.9	61.7 / 56.9	67.3 / 76.6	57.9 / 63.8	72.8 / 68.1	73.8 / 72.3	75.6 / 76.2	

Table 1: Transfer learning on video tracking tasks. All methods use the ResNet-50 backbone with the best results in bold.

Method	VOC 07 det		COCO det		COCO seg				
	AP ₅₀	AP ₇₅	AP ₅₀	AP ₇₅	AP ₅₀ ^m	AP ^m	AP ₇₅ ^m		
Supervised	74.4	42.4	42.7	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR	75.9	46.8	50.1	57.7	37.9	40.9	54.6	33.3	35.3
MoCo	77.1	46.8	52.5	58.9	39.3	42.5	55.8	34.4	36.5
BYOL	77.1	47.0	49.9	57.8	37.9	40.9	54.3	33.2	35.0
MAE	77.6	48.5	51.2	59.8	38.2	41.6	56.7	33.9	37.1
SimSiam	77.3	48.5	52.5	59.3	39.2	42.1	56.0	34.4	36.7
Barlow Twins	75.7	47.2	50.3	59.0	39.2	42.5	56.0	34.3	36.5
SwAV	75.5	46.5	49.6	58.6	38.4	41.3	55.2	33.8	35.9
VICRegL	75.9	47.4	52.3	59.2	39.8	42.1	56.5	35.1	36.8
SimCLR+TC ²	78.3	49.3	52.3	59.6	40.7	43.7	56.8	36.5	37.2
BYOL+TC ²	79.7	50.7	52.5	59.7	39.9	44.1	56.3	35.3	38.3
MAE+TC ²	81.2	50.6	53.4	62.3	40.5	44.7	59.2	35.1	40.0
VICRegL+TC ²	78.2	49.5	54.1	62.7	42.6	45.3	58.9	37.9	39.7

Table 2: Transfer learning on objection detection and instance segmentation. ‘‘AP’’ is the average precision, ‘‘AP_N’’ represents the ‘‘AP_N’’ with the IoU threshold $N\%$.

Ablation Studies

In this subsection, we conduct ablation studies to analyze how our proposed method perform well.

Loss function. Our loss function contains two important components: \mathcal{L}_v , \mathcal{L}_w . We reduce the correlation between generative factors by minimizing \mathcal{L}_v so that they can represent different semantic information. We also enforce correspondence between tasks and their generative factors by minimizing \mathcal{L}_w , while constraining the sparsity and diversity of the factors. Experimental results demonstrate the necessity of each component, as shown in Table 12.

Bi-level mechanism. Here, we explore the impact of bi-level mechanism, as shown in Table 12. Experimental results show that the bi-level optimization mechanism indeed improves the performance of the model.

The number of tasks n_t . Here, we conduct experiments to analyze the impact of the number of tasks n_t on the experimental results on the CIFAR10 dataset. In the experiment, we set the number of tasks to 2, 4, 8, 16, and 32. Table 13 shows the effect of the number of tasks on the experiments. The results indicate that $n_t = 4$ yields the best performance, which is also the setting we adopt.

Method	Photo	Sketch	Cartoon	Art	Average
SimCLR	86.4	85.1	87.2	74.3	83.3
SimCLR+TC ²	88.8	89.2	88.5	78.3	86.2
BYOL	83.9	84.6	82.7	64.5	78.9
BYOL+TC ²	85.6	87.2	85.0	69.9	81.9
Barlow Twins	83.1	82.7	83.4	64.8	78.5
Barlow Twins+TC ²	86.5	84.6	86.7	67.4	81.3
MAE	86.9	87.2	87.6	75.4	84.3
MAE+TC ²	89.1	90.0	89.2	78.9	86.8
SwAV	87.2	85.2	87.6	74.9	83.7
SwAV+TC ²	89.9	88.6	89.1	78.2	86.5
VICRegL	88.0	86.1	87.9	75.2	84.3
VICRegL+TC ²	91.5	89.3	90.2	79.6	87.7

Table 3: Transfer learning on PACS dataset.

Method	Average Accuracy(%)	Worst Accuracy (%)
SimCLR	85.2	82.4
SimCLR+TC ²	88.9	87.0
BYOL	85.9	83.1
BYOL+TC ²	89.4	87.8
MAE	86.8	84.0
MAE+TC ²	90.1	88.3

Table 4: Transfer learning on ColoredMNIST dataset.

Conclusion

This paper aims to explore how to improve the transferability of SSL. Through both theoretical analysis and empirical studies, we find that incorporating task-level information into SSL improves its transferability. However, task conflict introduces instability during the training process, limiting the transferability. To address this issue, we propose TC² to alleviate the negative impact of task conflict in self-supervised learning. TC² incorporates task-level information, calibrates task-factor correspondence through dedicated extraction functions, and optimizes the model using a two-stage bi-level optimization mechanism. Experimental results on various downstream tasks confirm the effectiveness of TC² in improving the transferability of SSL.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China under Grants 62506355 and 42506186, and the numerical calculations in this study were partially performed on the ORISE Supercomputer (DFZX202416).

References

- Albelwi, S. 2022. Survey on self-supervised learning: auxiliary pretext tasks and contrastive learning methods in imaging. *Entropy*, 24(4): 551.
- Bao, H.; Dong, L.; Piao, S.; and Wei, F. 2021. BEiT: BERT Pre-Training of Image Transformers. In *International Conference on Learning Representations*.
- Bardes, A.; Ponce, J.; and Lecun, Y. 2022. VICReg: Variance-Invariance-Covariance Regularization For Self-Supervised Learning. In *ICLR 2022-International Conference on Learning Representations*.
- Bousquet, O.; and Elisseeff, A. 2002. Stability and generalization. *The Journal of Machine Learning Research*, 2: 499–526.
- Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33: 9912–9924.
- Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.
- Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15750–15758.
- Deng, W.; Gould, S.; and Zheng, L. 2022. On the strong correlation between model invariance and generalization. *Advances in Neural Information Processing Systems*, 35: 28052–28067.
- Deshpande, S.; Wang, K.; Sreenivas, D.; Li, Z.; and Kuleshov, V. 2022. Deep multi-modal structural equations for causal effect estimation with unstructured proxies. *Advances in Neural Information Processing Systems*, 35: 10931–10944.
- Djulonga, J.; Yung, J.; Tschannen, M.; Romijnders, R.; Beyer, L.; Kolesnikov, A.; Puigcerver, J.; Minderer, M.; D’Amour, A.; Moldovan, D.; et al. 2020. On Robustness and Transferability of Convolutional Neural Networks. *arXiv preprint arXiv:2007.08558*.
- Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, 1422–1430.
- Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338.
- Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.
- Gat, I.; Schwartz, I.; Schwing, A.; and Hazan, T. 2020. Removing bias in multi-modal classifiers: Regularization by maximizing functional entropies. *Advances in Neural Information Processing Systems*, 33: 3197–3208.
- Gordon, J.; Bronskill, J.; Bauer, M.; Nowozin, S.; and Turner, R. E. 2018. Meta-learning probabilistic inference for prediction. *arXiv preprint arXiv:1805.09921*.
- Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent—a new approach to self-supervised learning. *Advances in neural information processing systems*, 33: 21271–21284.
- Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 297–304. JMLR Workshop and Conference Proceedings.
- He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.
- He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.
- Hospedales, T.; Antoniou, A.; Micaelli, P.; and Storkey, A. 2021. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5149–5169.
- Hu, Z.; Zhao, Z.; Yi, X.; Yao, T.; Hong, L.; Sun, Y.; and Chi, E. 2022. Improving multi-task generalization via regularizing spurious correlation. *Advances in Neural Information Processing Systems*, 35: 11450–11466.
- Jiang, Y.; Chen, Z.; Kuang, K.; Yuan, L.; Ye, X.; Wang, Z.; Wu, F.; and Wei, Y. 2022. The Role of Deconfounding in Meta-learning. In *International Conference on Machine Learning*, 10161–10176. PMLR.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, 740–755. Springer.
- Liu, X.; Wang, Z.; Li, Y.-L.; and Wang, S. 2022a. Self-supervised learning via maximum entropy coding. *Advances in Neural Information Processing Systems*, 35: 34091–34105.
- Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Philip, S. Y. 2022b. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 5879–5900.

- Mignacco, F.; Krzakala, F.; Lu, Y.; Urbani, P.; and Zdeborova, L. 2020. The role of regularization in classification of high-dimensional noisy gaussian mixture. In *International conference on machine learning*, 6874–6883. PMLR.
- Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; and Schindler, K. 2016. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.
- Nam, J.; Cha, H.; Ahn, S.; Lee, J.; and Shin, J. 2020. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33: 20673–20684.
- Ni, R.; Shu, M.; Souri, H.; Goldblum, M.; and Goldstein, T. 2021. The close relationship between contrastive learning and meta-learning. In *International conference on learning representations*.
- Pan, S. J.; and Yang, Q. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359.
- Pearl, J. 2009. *Causality*. Cambridge university press.
- Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Van Gool, L.; Gross, M.; and Sorkine-Hornung, A. 2016. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 724–732.
- Schiappa, M. C.; Rawat, Y. S.; and Shah, M. 2023. Self-supervised learning for videos: A survey. *ACM Computing Surveys*, 55(13s): 1–37.
- Schölkopf, B.; Locatello, F.; Bauer, S.; Ke, N. R.; Kalchbrenner, N.; Goyal, A.; and Bengio, Y. 2021. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634.
- Standley, T.; Zamir, A.; Chen, D.; Guibas, L.; Malik, J.; and Savarese, S. 2020. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, 9120–9132. PMLR.
- Sun, C.; He, P.; Ji, Q.; Zang, Z.; Li, J.; Wang, R.; and Wang, W. 2024a. M2i2: Learning efficient multi-agent communication via masked state modeling and intention inference. *arXiv preprint arXiv:2501.00312*.
- Sun, C.; He, P.; Wang, R.; and Zheng, C. 2025. Revisiting Communication Efficiency in Multi-Agent Reinforcement Learning from the Dimensional Analysis Perspective. In Das, S.; Nowé, A.; and Vorobeychik, Y., eds., *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025, Detroit, MI, USA, May 19-23, 2025*, 1977–1986. International Foundation for Autonomous Agents and Multiagent Systems / ACM.
- Sun, C.; Wu, B.; Wang, R.; Hu, X.; Yang, X.; and Cong, C. 2021. Intrinsic Motivated Multi-Agent Communication. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '21*, 1668–1670. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.
- Sun, C.; Zang, Z.; Li, J.; Li, J.; Xu, X.; Wang, R.; and Zheng, C. 2024b. T2mac: Targeted and trusted multi-agent communication through selective engagement and evidence-driven integration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15154–15163.
- Suter, R.; Miladinovic, D.; Schölkopf, B.; and Bauer, S. 2019. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, 6056–6065. PMLR.
- Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5018–5027.
- Vilalta, R.; and Drissi, Y. 2002. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18: 77–95.
- Wang, T.; and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, 9929–9939. PMLR.
- Wang, Z.; Zhao, H.; Li, Y.-L.; Wang, S.; Torr, P.; and Bertinetto, L. 2021. Do different tracking tasks require different appearance models? *Advances in Neural Information Processing Systems*, 34: 726–738.
- Wolff, P.; and Zettergren, M. 2019. A vector model of causal meaning. In *Proceedings of the twenty-fourth annual conference of the Cognitive Science Society*, 944–949. Routledge.
- Xu, J.; Xiao, L.; and López, A. M. 2019. Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 7: 156694–156706.
- Ying, W.; Zhang, Y.; Huang, J.; and Yang, Q. 2018. Transfer learning via learning to transfer. In *International Conference on Machine Learning*, 5085–5094. PMLR.
- Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27.
- Zamir, A. R.; Sax, A.; Shen, W.; Guibas, L. J.; Malik, J.; and Savarese, S. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3712–3722.
- Zang, Z.; Sun, C.; Liu, L.; Sun, F.; and Zheng, C. 2025. Loss of Plasticity: A New Perspective on Solving Multi-Agent Exploration for Sparse Reward Tasks. In Das, S.; Nowé, A.; and Vorobeychik, Y., eds., *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025, Detroit, MI, USA, May 19-23, 2025*, 2299–2308. International Foundation for Autonomous Agents and Multiagent Systems / ACM.
- Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, 12310–12320. PMLR.
- Zhu, S.; An, B.; and Huang, F. 2021. Understanding the generalization benefit of model invariance from a data perspective. *Advances in Neural Information Processing Systems*, 34: 4328–4341.