ON THE COUNTERFACTUAL RELATION CERTIFICATION OF SELF-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised learning (SSL) learns representations by constructing supervision from unlabeled data. However, satisfying self-supervised objectives does not guarantee reliable representations, since the same training relations may be explained by semantic factors or by nuisance cues such as background. Consequently, an SSL model may achieve low training loss while relying on spurious dependencies that fail under distribution, task, or semantic shifts. We address this issue from a counterfactual perspective and characterize model reliability through two complementary principles: the learned dependency should be revised when essential semantic factors change, while remaining stable when semantic factors are preserved under nuisance variations. Since true counterfactual interventions are unavailable in SSL, we propose General SSL (GeSSL), which approximates these principles through observable training proxies. GeSSL models counterfactual necessity with a discriminative relation loss and counterfactual sufficiency through consistency at the sample, support-query set, and task-distribution levels. Theoretical analysis establishes bounded generalization error on novel tasks. Extensive experiments show that GeSSL consistently improves representative discriminative and generative SSL methods across multiple evaluation protocols.

1 INTRODUCTION

Self-supervised learning (SSL) has become a fundamental paradigm for representation learning from unlabeled data (Gui et al., 2024). Existing methods can be broadly categorized into discriminative SSL (D-SSL) and generative SSL (G-SSL). D-SSL methods, such as SimCLR (Chen et al., 2020a), learn representations by enforcing consistency across augmented views, while G-SSL methods, such as MAE (He et al., 2022), learn by reconstructing missing or corrupted content. Despite their different forms, both paradigms derive supervision from relations constructed directly from unlabeled data.

However, satisfying these self-supervised relations does not guarantee that the model learns reliable representations. The same relation can often be explained by either semantic factors or nuisance cues. For example, two augmented views may be matched because they preserve object semantics, but also because they share background, texture, or augmentation-specific artifacts. As a result, an SSL model may achieve a low training loss while relying on spurious dependencies that fail under distribution or task shifts. The central problem is therefore not whether the constructed relations are valid, but whether the *model learns them for the right reasons*.

We study this problem from a counterfactual perspective. A reliable SSL model should respond differently to two types of hypothetical changes. If semantic factors change while nuisance factors remain similar, the learned relation should be revised; otherwise, the model may rely on superficial similarity. Conversely, if semantic factors are preserved while nuisance factors change, the learned relation should remain stable. Inspired by the counterfactual notions of necessity and sufficiency (Pearl, 2009; Pearl et al., 2016), we refer to these two properties as *counterfactual necessity* and *counterfactual sufficiency*. We use these notions as principles for model reliability rather than claiming causal identification from unlabeled observational data.

Based on this perspective, we propose **General SSL (GeSSL)**, a unified framework that approximates the two counterfactual principles through observable training proxies. For counterfactual necessity, GeSSL introduces a discriminative relation loss that learns an adaptive boundary around each anchor in the representation space, pulling compatible samples closer while separating incompatible ones.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

This encourages the model to revise learned dependencies when informative factors change, rather than relying on superficial similarity. For counterfactual sufficiency, GeSSL promotes the stability of learned dependencies at three levels. At the sample level, different augmentations of the same sample are aligned to improve robustness to local nuisance perturbations. At the set level, support-query bi-level optimization evaluates whether dependencies learned from one subset remain valid on another, reducing overfitting to specific sample compositions. At the distribution level, multiple mini-batches are treated as sampled tasks, encouraging the learned dependency to remain stable across broader task variations. Together, these objectives encourage the model to be sensitive to semantic changes while remaining robust to nuisance variations. We further provide theoretical analysis showing bounded generalization error on unseen tasks, and extensive experiments demonstrate consistent improvements over representative D-SSL and G-SSL baselines across multiple evaluation protocols.

Our contributions are summarized as follows. **(i)** We identify a fundamental reliability issue in SSL: models may satisfy self-supervised objectives through spurious dependencies rather than semantically meaningful factors. **(ii)** We introduce a counterfactual view of SSL reliability based on two model-level principles, counterfactual necessity and counterfactual sufficiency. **(iii)** We propose GeSSL, which approximates these principles through a discriminative relation objective and sample-, set-, and distribution-level consistency. **(iv)** We provide theoretical analysis and extensive experiments demonstrating consistent improvements over representative D-SSL and G-SSL baselines.

2 Revisiting SSL from a Relation Perspective

During training, unlabeled data are usually organized into mini-batches. Let a mini-batch be denoted as $X_{tr} = \{x_i\}_{i=1}^{N}$, where x_i is the i-th sample and N is the batch size. In D-SSL, each sample x_i is transformed into two augmented views, denoted as x_i^1 and x_i^2. In G-SSL, such as masked image modeling, a corrupted view x_i^1 is constructed from x_i, while the original or unmasked sample is used as the reconstruction target x_i^2. Thus, both D-SSL and G-SSL can be written in a unified form as $X_{tr}^{aug} = \{(x_i^1, x_i^2)\}_{i=1}^{N}$, where each pair provides a self-supervised learning signal.

From this view, SSL does not learn from human-provided labels, but from relations constructed by the training objective itself. In D-SSL, the relation is defined between two augmented views of the same instance, and the model is encouraged to make their representations consistent, often with additional regularization or contrastive separation (Chen et al., 2020a; Oord et al., 2018; Hjelm et al., 2018). In G-SSL, the relation is defined between a corrupted input and its reconstruction target, and the model is trained to recover x_i^2 from x_i^1 (He et al., 2022). Therefore, despite their different objective forms, D-SSL and G-SSL both rely on self-constructed relations as surrogate supervision. To make this relation explicit, we denote one view in each pair as an anchor x_i^{anc} and the other as its paired view x_i^v. The SSL objective assumes that x_i^v should be related to x_i^{anc}, which can be written as $r_i : x_i^v \sim x_i^{\mathrm{anc}}$. A mini-batch then induces a set of candidate self-supervised relations, $\mathcal{R}(X_{tr}^{aug}) = \{r_i\}_{i=1}^{N}$, which serves as the effective supervision.

However, satisfying these self-supervised relations does not necessarily lead to reliable representations. The same relation can often be learned through different underlying factors. For example, two views of the same instance may be matched because they preserve task-relevant semantics, but the model may also satisfy the same objective by exploiting nuisance cues such as background, texture, or augmentation artifacts. Similarly, separating different instances does not guarantee that the learned distinction reflects semantically meaningful factors. Therefore, the ambiguity lies not in the constructed relations themselves, but in *how the model learns to satisfy them*. A mini-batch can thus be viewed as a local relation environment in which the observed relations provide supervision, while the underlying dependencies learned by the model remain uncertain. This perspective motivates us to examine whether these learned dependencies behave appropriately under counterfactual changes. In the next section, we introduce counterfactual necessity and counterfactual sufficiency as two complementary principles for characterizing the reliability of self-supervised models.

3 Methodology

In this section, we present GeSSL. We first introduce two counterfactual principles for characterizing the reliability of self-supervised models, and then describe how GeSSL approximates them using

observable training signals in the absence of true counterfactual interventions. The overall framework is illustrated in Figure 1.

3.1 Counterfactual Principles for Reliable SSL

To characterize the desired behavior of a reliable model, we conceptually decompose an observation into semantic factors Z_c and nuisance factors Z_s. Here, Z_c denotes factors that determine task-relevant semantic content, while Z_s denotes non-essential variations such as background, texture, style, viewpoint, and occlusion. We do not assume that Z_c or Z_s are observed, disentangled, or directly intervenable in SSL. Instead, they are introduced only to describe how the learned model should respond to hypothetical changes in the underlying factors. Inspired by the counterfactual notions of necessity and sufficiency in causal reasoning (Pearl, 2009; Pearl et al., 2016), we introduce two complementary principles for characterizing model reliability.

Counterfactual necessity. Consider a hypothetical change in which semantic factors vary while nuisance factors remain similar. If the model still preserves the same learned relation, its behavior may be explained by nuisance similarity rather than by the semantic factors that should distinguish the observations. A reliable model should therefore revise the learned dependency when the semantic evidence supporting the relation changes. Counterfactual necessity thus requires the learned representation to remain sensitive to informative semantic changes, preventing superficially similar but semantically incompatible observations from being treated as equivalent.

Counterfactual sufficiency. Conversely, consider a hypothetical change in which semantic factors are preserved while nuisance factors vary. If the learned relation changes substantially under such variations, the model is likely to rely on unstable nuisance information. A reliable model should therefore preserve the learned dependency when semantic content remains unchanged despite nuisance perturbations. Counterfactual sufficiency thus requires the learned representation to remain stable under non-essential variations that should not alter its semantic behavior.

These two principles impose complementary requirements on SSL: counterfactual necessity encourages sensitivity to semantic changes, whereas counterfactual sufficiency encourages invariance to nuisance changes. However, true interventions on Z_c and Z_s are unavailable in standard SSL, so these properties cannot be directly enforced. GeSSL therefore approximates them through observable training proxies. For counterfactual necessity, we introduce a discriminative relation loss that learns an adaptive compatibility boundary around each anchor, encouraging the model to distinguish compatible from incompatible observations. For counterfactual sufficiency, we evaluate the stability of learned dependencies at three levels: sample-level consistency across augmented views, set-level consistency through support-query bi-level optimization, and distribution-level consistency across mini-batches sampled from the empirical task distribution.

3.2 The Proposed Method: GeSSL

Let the augmented mini-batches be denoted by $X^{\text{aug}} = \{X^{\text{aug}}_{tr,l}\}_{l=1}^{m}$, where l indexes the l-th mini-batch and m is the number of mini-batches processed in parallel. The l-th mini-batch is written as $X^{\text{aug}}_{tr,l} = \{(x^1_{i,l}, x^2_{i,l}, x^{\text{anc}}_{i,l})\}_{i=1}^{n}$, where $x^{\text{anc}}_{i,l}$ is the anchor, and $x^1_{i,l}$ and $x^2_{i,l}$ are two perturbed views associated with the same anchor. For D-SSL, the three views can be generated from the same source image through different augmentations. For G-SSL, the anchor can be the original target sample, while the paired views are generated by different masking or corruption operations.

Each triplet induces two relation sets: the support relation set $X^{\text{sup}}_{tr,l} = \{(x^1_{i,l}, x^{\text{anc}}_{i,l})\}_{i=1}^{n}$ and the query relation set $X^{\text{que}}_{tr,l} = \{(x^2_{i,l}, x^{\text{anc}}_{i,l})\}_{i=1}^{n}$. The support relations provide one perturbed realization for learning, while the query relations provide another perturbed realization for testing whether the learned relations remain stable. GeSSL is formulated as

$$\min_{f,g} \textstyle\sum_{l=1}^{m} \mathcal{L}_{ssl}(f', X^{\text{que}}_{tr,l}), \text{s.t.} f' = \arg\min_{\tilde{f}} \textstyle\sum_{l=1}^{m} [\mathcal{L}_{ssl}(\tilde{f}, X^{\text{sup}}_{tr,l}) + \mu \mathcal{L}_{disc}(\tilde{f}, g, X^{\text{sup}}_{tr,l})], \quad (1)$$

where f is the feature extractor, g is an auxiliary network for relation boundary prediction, $\mathcal{L}_{ssl}$ is the base SSL loss, and μ is a balancing hyperparameter. For D-SSL, $\mathcal{L}_{ssl}$ can be instantiated as a contrastive or alignment-based loss; for G-SSL, it can be instantiated as a reconstruction loss.

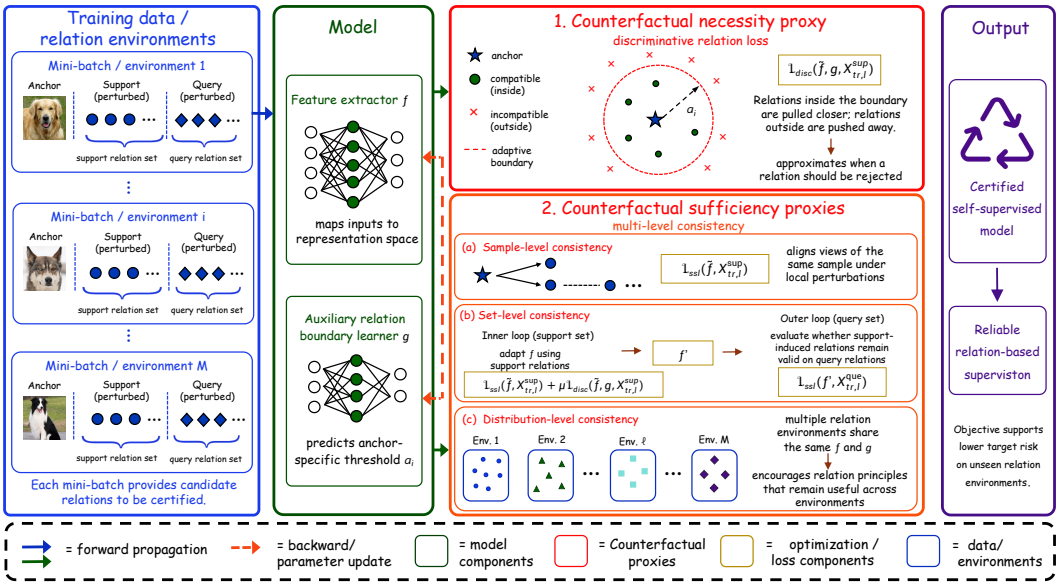


Figure 1: Overview of GeSSL. Self-supervised pairs are treated as candidate relations. The discriminative relation loss approximates counterfactual necessity by learning an adaptive relation boundary. Sample-level alignment, support-query bi-level optimization, and multi-batch training approximate counterfactual sufficiency at different levels.

3.3 Approximating Counterfactual Necessity

Counterfactual necessity requires the model to revise its learned dependency when semantic compatibility changes. Since semantic factors are unobserved in SSL, GeSSL approximates this principle with an adaptive relation boundary around each anchor, encouraging sensitivity to informative changes in the representation space.

For the i-th anchor in the l-th mini-batch, we compare it with all support views in the same mini-batch. Let $d^i_j = d(f(x^1_{j,l}), f(x^{\text{anc}}_{i,l}))$, where $d(\cdot,\cdot)$ denotes cosine distance. The auxiliary network g predicts an anchor-specific threshold $a_i \in \mathbb{R}$ from the statistics of relative representations:

$$a_i = g(\text{Mean}\{f(x^1_{j,l}) - f(x^{\text{anc}}_{i,l})\}^n_{j=1}, \text{Cov}\{f(x^1_{j,l}) - f(x^{\text{anc}}_{i,l})\}^n_{j=1}).$$

Intuitively, samples inside the boundary are treated as compatible with the anchor and are pulled closer, while samples outside the boundary are treated as incompatible and are pushed away. The corresponding hard relation loss is

$$\mathcal{L}_{disc}(f, g, X^{\text{sup}}_{tr,l}) = \sum^n_{i=1}\sum^n_{j=1}[\mathbb{1}_{\{d^i_j \le a_i\}} \cdot d^i_j + \mathbb{1}_{\{d^i_j > a_i\}} \cdot (-d^i_j)]. \tag{2}$$

Minimizing this loss decreases the distance for compatible pairs and increases the distance for incompatible pairs. Because the indicator function is non-differentiable, we use a sigmoid relaxation:

$$\mathcal{L}_{disc}(f, g, X^{\text{sup}}_{tr,l}) = \sum^n_{i=1}\sum^n_{j=1}[w^i_j \cdot d^i_j + (1 - w^i_j) \cdot (-d^i_j)], \tag{3}$$

where $w^i_j = \text{Sigmoid}\left(k(a_i - d^i_j)\right)$, and $k \in \mathbb{R}^+$ controls the sharpness of the boundary.

This loss is not intended to recover true semantic classes. Instead, it serves as a trainable proxy for counterfactual necessity by encouraging the model to revise learned dependencies when the evidence for compatibility becomes insufficient. The boundary learner g is optimized through the bi-level objective, such that its effectiveness is evaluated by whether the induced support update improves consistency on the query set. In this way, GeSSL discourages indiscriminate relation preservation and improves the discriminative sensitivity of the learned representation without requiring labels.

3.4 APPROXIMATING COUNTERFACTUAL SUFFICIENCY

Counterfactual sufficiency requires the model to preserve its learned dependency when semantic content remains unchanged despite nuisance variations. Since true interventions on nuisance factors are unavailable, GeSSL approximates this behavior through three observable levels of consistency.

Sample-level consistency. At the sample level, GeSSL follows the standard SSL assumption that different augmentations or corruptions of the same sample preserve the main semantic content. The base loss $\mathcal{L}_{ssl}$ aligns each paired view with its anchor, encouraging the relation to remain stable under local perturbations. This provides the most basic proxy for counterfactual sufficiency.

Set-level consistency. At the set level, GeSSL tests whether relations learned from one perturbed relation set still hold on another. The support relations are used to obtain a proxy model f', while the query relations evaluate whether the support-induced update generalizes beyond the support set. The inner-loop update is

$$f'(f,g) \leftarrow f - \alpha \nabla_f \textstyle\sum_{l=1}^{m} [\mathcal{L}_{ssl}(f, X_{tr,l}^{\text{sup}}) + \mu \mathcal{L}_{disc}(f, g, X_{tr,l}^{\text{sup}})], \tag{4}$$

where α is the inner-loop learning rate. In practice, we use one inner-loop step for efficiency. The outer loop updates f and g according to the query loss:

$$f \leftarrow f - \beta \textstyle\sum_{l=1}^{m} \nabla_f \mathcal{L}_{ssl}(f'(f,g), X_{tr,l}^{\text{que}}), \quad g \leftarrow g - \beta \textstyle\sum_{l=1}^{m} \nabla_g \mathcal{L}_{ssl}(f'(f,g), X_{tr,l}^{\text{que}}), \tag{5}$$

where β is the learning rate. Since f' depends on both f and g, the outer loop updates the representation and the relation boundary according to query performance. Thus, relations that only fit the support perturbation are discouraged, while relations that remain valid on query views are reinforced.

Distribution-level consistency. At the distribution level, GeSSL processes m mini-batches in parallel and treats them as local relation environments sampled from an empirical task distribution. The same feature extractor f and relation boundary learner g are shared across these environments. This encourages relation principles that are not specific to a single mini-batch, but remain useful across multiple relation environments. As a result, GeSSL promotes representations that are more likely to transfer to unseen tasks.

Together, sample-level consistency addresses local perturbations, set-level consistency addresses support-query shifts, and distribution-level consistency addresses broader task-environment variations. These three levels provide observable proxies for encouraging the learned dependencies to remain stable under progressively broader forms of nuisance variation.

3.5 FURTHER DISCUSSION

GeSSL can be understood as a trainable approximation to counterfactual model reliability. The self-supervised relations provide the training constraints, while GeSSL regularizes how the model learns to satisfy them. Specifically, the discriminative relation loss approximates counterfactual necessity by encouraging the model to revise learned dependencies when compatibility evidence changes, whereas the three-level consistency design approximates counterfactual sufficiency by encouraging these dependencies to remain stable under local perturbations, support-query shifts, and broader task-environment variations. GeSSL does not attempt to recover true semantic factors, identify causal effects, or generate genuine counterfactual samples. Instead, it uses the counterfactual logic of necessity and sufficiency as model-level reliability principles and realizes them through observable self-supervised training signals.

GeSSL is also related to meta-learning through its support-query bi-level optimization, but the role of this structure is fundamentally different. Conventional meta-learning typically aims to learn representations or initialization strategies that can rapidly adapt to new tasks, often with task-level supervision. In GeSSL, the support-query split is not introduced to learn task-specific classifiers or to optimize fast adaptation. Instead, it provides an update-then-test mechanism: the support set induces a model update, and the query set evaluates whether the learned dependency remains stable beyond the particular relations used for that update. The resulting bi-level optimization therefore serves as a robustness test on the learned self-supervised dependency, rather than as a direct instance of supervised meta-learning. This distinction is central to GeSSL: episodic optimization is used as an observable proxy for testing the transferability of learned dependencies, while the ultimate objective remains reliable self-supervised representation learning.

4 THEORETICAL ANALYSIS

In this section, we analyze the generalization behavior of GeSSL. Our goal is not to prove recovery of true semantic factors or genuine counterfactual effects, which are unobserved in SSL. Instead, we first study whether the observable objective optimized by GeSSL generalizes from the training episodes to unseen self-supervised episodes. We then discuss when such generalization can further imply control of downstream target risk.

Self-supervised episodes. Let $\tau \sim \mathcal{T}$ denote a self-supervised episode sampled from an underlying episode distribution $\mathcal{T}$. Each episode $\tau = (S_\tau, Q_\tau)$ contains a support set S_τ and a query set Q_τ. The support set is used to form the inner-loop update, whereas the query set evaluates whether the dependency learned from the support observations remains stable beyond the samples used for that update. Let f_θ denote the representation encoder and g_ϕ the adaptive relation-boundary learner. Given S_τ, the updated representation parameter is $\theta'_\tau = A(\theta, \phi, S_\tau)$, where $A(\cdot)$ denotes the inner-loop update defined in Eq. equation 4. For each episode, we define the observable GeSSL objective as

$$\mathcal{J}(\theta, \phi; \tau) = \mathcal{L}_{ssl}(\theta; S_\tau) + \mu\overline{\mathcal{L}}_{disc}(\theta, \phi; S_\tau) + \mathcal{L}_{query}(\theta'_\tau; Q_\tau), \tag{6}$$

where $\mathcal{L}_{ssl}$ is the base SSL loss, $\overline{\mathcal{L}}_{disc}$ is a non-negative version of the discriminative relation loss used only for analysis, and $\mathcal{L}_{query}$ is the SSL loss evaluated on the query set after the support update. If $\mathcal{L}_{disc}$ is lower bounded by a constant c_{disc} on the considered parameter domain, we define $\overline{\mathcal{L}}_{disc} = \mathcal{L}_{disc} - c_{disc}$; this constant shift does not affect optimization. The three terms correspond to sample-level consistency, the observable proxy for counterfactual necessity, and set-level consistency, respectively. Distribution-level consistency is induced by optimizing the same objective across multiple episodes sampled from $\mathcal{T}$. Let $\mathcal{G} = \{\tau \mapsto \mathcal{J}(\theta, \phi; \tau) : (\theta, \phi) \in \Theta \times \Phi\}$ denote the function class induced by GeSSL. We assume that $0 \le \mathcal{J}(\theta, \phi; \tau) \le B$ for all (θ, ϕ) and τ. Let $\mathfrak{R}_M(\mathcal{G})$ denote the expected Rademacher complexity of $\mathcal{G}$ over M sampled episodes.

Theorem 4.1. *Let $\{\tau_i\}_{i=1}^M$ be M episodes sampled i.i.d. from $\mathcal{T}$. Then, with probability at least $1 - \delta$, the following inequality holds simultaneously for all (θ, ϕ):*

$$\mathbb{E}_{\tau\sim\mathcal{T}}\left[\mathcal{J}(\theta, \phi; \tau)\right] \le \frac{1}{M}\sum_{i=1}^{M}\mathcal{J}(\theta, \phi; \tau_i) + 2\mathfrak{R}_M(\mathcal{G}) + B\sqrt{\frac{\log(1/\delta)}{2M}}. \tag{7}$$

The bound also holds for the data-dependent parameters $(\hat{\theta}, \hat{\phi})$ obtained by GeSSL training.

The proof is provided in Appendix A. Theorem 4.1 shows that minimizing the empirical GeSSL objective over multiple training episodes controls its expected value on unseen episodes, up to a hypothesis-complexity term and a finite-sample term. This result provides the theoretical interpretation of distribution-level consistency in GeSSL: the model is not optimized for a single support-query split, but for an objective shared across episodes drawn from the same underlying distribution. Importantly, Theorem 4.1 concerns the observable self-supervised objective and does not by itself imply recovery of downstream semantics. To explicitly characterize this additional connection, we introduce the following task-dependent assumption.

Assumption 4.2. *Let $\mathcal{L}_{tar}(A(\theta, \phi, S_\tau); \tau)$ denote a downstream target loss that is not observed during SSL training. There exist constants $\alpha > 0$ and $\varepsilon_{\text{proxy}} \ge 0$ such that*

$$\mathbb{E}_{\tau\sim\mathcal{T}}\left[\mathcal{L}_{tar}\big(A(\theta, \phi, S_\tau); \tau\big)\right] \le \alpha\mathbb{E}_{\tau\sim\mathcal{T}}\left[\mathcal{J}(\theta, \phi; \tau)\right] + \varepsilon_{\text{proxy}}. \tag{8}$$

The term $\varepsilon_{\text{proxy}}$ explicitly captures the mismatch between the observable self-supervised signals and the unobserved downstream semantics. This assumption is not a claim that SSL supervision uniquely identifies semantic factors; rather, it states the additional condition required for improvements in the observable GeSSL objective to translate into downstream performance. Combining Assumption 4.2 with Theorem 4.1 immediately gives the following result.

Corollary 4.3. *Under Assumption 4.2, with probability at least $1 - \delta$, we have the following:*

$$\mathbb{E}_{\tau\sim\mathcal{T}}\left[\mathcal{L}_{tar}\big(A(\hat{\theta}, \hat{\phi}, S_\tau); \tau\big)\right] \le \alpha\left[\frac{1}{M}\sum_{i=1}^{M}\mathcal{J}(\hat{\theta}, \hat{\phi}; \tau_i) + 2\mathfrak{R}_M(\mathcal{G}) + B\sqrt{\frac{\log(1/\delta)}{2M}}\right] + \varepsilon_{\text{proxy}}. \tag{9}$$

Table 1: The Top-1 and Top-5 classification accuracies of linear classifier on the ImageNet-100 and ImageNet (200 Epochs) with ResNet-50.

Method	ImageNet-100		ImageNet	
	Top-1	Top-5	Top-1	Top-5
SimCLR (Chen et al., 2020a)	70.15 ± 0.16	89.75 ± 0.14	68.32 ± 0.31	89.76 ± 0.23
MoCo (Chen et al., 2020b)	72.80 ± 0.12	91.64 ± 0.11	67.55 ± 0.27	88.42 ± 0.11
SimSiam (Chen & He, 2021)	73.01 ± 0.21	92.61 ± 0.27	70.02 ± 0.14	88.76 ± 0.23
Barlow Twins (Zbontar et al., 2021)	75.97 ± 0.23	92.91 ± 0.19	69.94 ± 0.32	88.97 ± 0.27
MAE (He et al., 2022)	76.56 ± 0.16	93.24 ± 0.24	70.73 ± 0.25	91.41 ± 0.27
DINO (Caron et al., 2021)	75.43 ± 0.18	93.32 ± 0.19	70.58 ± 0.24	91.32 ± 0.27
W-MSE (Ermolov et al., 2021)	76.01 ± 0.27	93.12 ± 0.21	70.85 ± 0.31	91.57 ± 0.20
RELIC v2 (Tomasev et al., 2022)	75.88 ± 0.15	93.52 ± 0.13	70.98 ± 0.21	91.15 ± 0.26
LMCL (Chen et al., 2021)	75.89 ± 0.19	92.89 ± 0.28	70.83 ± 0.26	90.04 ± 0.21
ReSSL (Zheng et al., 2021)	75.77 ± 0.21	92.91 ± 0.27	69.92 ± 0.24	91.25 ± 0.12
CorInfoMax (Ozsoy et al., 2022)	75.54 ± 0.20	92.23 ± 0.25	70.83 ± 0.15	91.53 ± 0.22
MEC (Liu et al., 2022a)	75.38 ± 0.17	92.84 ± 0.20	70.34 ± 0.27	91.25 ± 0.38
VICRegL (Bardes et al., 2022)	75.96 ± 0.19	92.97 ± 0.26	70.24 ± 0.27	91.60 ± 0.24
SimCLR + GeSSL	72.96 ± 0.24	92.50 ± 0.17	69.88 ± 0.21	91.32 ± 0.25
MoCo + GeSSL	74.35 ± 0.24	94.10 ± 0.31	69.60 ± 0.30	91.28 ± 0.39
SimSiam + GeSSL	75.93 ± 0.24	95.51 ± 0.38	72.04 ± 0.22	89.43 ± 0.40
Barlow Twins + GeSSL	77.55 ± 0.29	93.48 ± 0.30	72.84 ± 0.26	89.50 ± 0.19
MAE + GeSSL	78.45 ± 0.31	$\mathbf{96.17 \pm 0.26}$	71.45 ± 0.24	92.19 ± 0.30
DINO + GeSSL	77.13 ± 0.29	95.75 ± 0.30	73.52 ± 0.30	$\mathbf{94.05 \pm 0.26}$
VICRegL + GeSSL	$\mathbf{78.48 \pm 0.34}$	95.90 ± 0.17	$\mathbf{73.91 \pm 0.36}$	93.77 ± 0.35

Table 2: The semi-supervised learning accuracies ($\pm$ 95% confidence interval) on ImageNet with the ResNet-50 pre-trained on Imagenet.

Method	Epochs	1%		10%	
		Top-1	Top-5	Top-1	Top-5
MoCo (Chen et al., 2020b)	200	43.8 ± 0.2	72.3 ± 0.1	61.9 ± 0.1	84.6 ± 0.2
BYOL (Grill et al., 2020)	200	54.8 ± 0.2	78.8 ± 0.1	68.0 ± 0.2	88.5 ± 0.2
MoCo + GeSSL	200	46.6 ± 0.3	74.5 ± 0.3	63.8 ± 0.2	85.9 ± 0.2
BYOL + GeSSL	200	$\mathbf{57.3 \pm 0.2}$	$\mathbf{79.8 \pm 0.2}$	$\mathbf{71.1 \pm 0.2}$	$\mathbf{90.1 \pm 0.3}$
SimCLR (Chen et al., 2020a)	1000	48.3 ± 0.2	75.5 ± 0.1	65.6 ± 0.1	87.8 ± 0.2
MoCo (Chen et al., 2020b)	1000	52.3 ± 0.1	77.9 ± 0.2	68.4 ± 0.1	88.0 ± 0.2
BYOL (Grill et al., 2020)	1000	56.3 ± 0.2	79.6 ± 0.2	69.7 ± 0.2	89.3 ± 0.1
SimSiam (Chen & He, 2021)	1000	54.9 ± 0.2	79.5 ± 0.2	68.0 ± 0.1	89.0 ± 0.3
Barlow Twins (Zbontar et al., 2021)	1000	55.0 ± 0.1	79.2 ± 0.1	67.7 ± 0.2	89.3 ± 0.2
RELIC v2 (Tomasev et al., 2022)	1000	55.2 ± 0.2	80.0 ± 0.1	68.0 ± 0.2	88.9 ± 0.2
LMCL (Chen et al., 2021)	1000	54.8 ± 0.2	79.4 ± 0.2	70.3 ± 0.1	89.9 ± 0.2
ReSSL (Zheng et al., 2021)	1000	55.0 ± 0.1	79.6 ± 0.3	69.9 ± 0.1	89.7 ± 0.1
SSL-HSIC (Li et al., 2021)	1000	55.4 ± 0.3	80.1 ± 0.2	70.4 ± 0.1	90.0 ± 0.1
CorInfoMax (Ozsoy et al., 2022)	1000	55.0 ± 0.2	79.6 ± 0.3	70.3 ± 0.2	89.3 ± 0.2
MEC (Liu et al., 2022a)	1000	54.8 ± 0.1	79.4 ± 0.2	70.0 ± 0.1	89.1 ± 0.1
VICRegL (Bardes et al., 2022)	1000	54.9 ± 0.1	79.6 ± 0.2	67.2 ± 0.1	89.4 ± 0.2
SimCLR + GeSSL	1000	51.1 ± 0.2	77.7 ± 0.1	67.8 ± 0.3	89.8 ± 0.3
MoCo + GeSSL	1000	54.0 ± 0.3	78.8 ± 0.1	71.6 ± 0.2	89.3 ± 0.2
BYOL + GeSSL	1000	$\mathbf{59.6 \pm 0.3}$	$\mathbf{81.9 \pm 0.2}$	$\mathbf{71.8 \pm 0.2}$	91.3 ± 0.2
Barlow Twins + GeSSL	1000	58.1 ± 0.3	80.5 ± 0.2	68.9 ± 0.2	$\mathbf{92.3 \pm 0.3}$

Table 3: The results of transfer learning on object detection and instance segmentation with C4-backbone as the feature extractor. "AP" is the average precision, "AP_N" represents the average precision when the IoU (Intersection and Union Ratio) threshold is $N\%$.

Method	VOC 07 detection			VOC 07+12 detection			COCO detection			COCO instance segmentation		
	$\mathbf{AP_{50}}$	$\mathbf{AP}$	$\mathbf{AP_{75}}$	$\mathbf{AP_{50}}$	$\mathbf{AP}$	$\mathbf{AP_{75}}$	$\mathbf{AP_{50}}$	$\mathbf{AP}$	$\mathbf{AP_{75}}$	$\mathbf{AP^{mask}_{50}}$	$\mathbf{AP^{mask}}$	$\mathbf{AP^{mask}_{75}}$
Supervised	74.4	42.4	42.7	81.3	53.5	58.8	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR (Chen et al., 2020a)	75.9	46.8	50.1	81.8	55.5	61.4	57.7	37.9	40.9	54.6	33.3	35.3
MoCo (Chen et al., 2020b)	77.1	46.8	52.5	82.5	57.4	64.0	58.9	39.3	42.5	55.8	34.4	36.5
BYOL (Grill et al., 2020)	77.1	47.0	49.9	81.4	55.3	61.1	57.8	37.9	40.9	54.3	33.2	35.0
SimSiam (Chen & He, 2021)	77.3	48.5	52.5	82.4	57.0	63.7	59.3	39.2	42.1	56.0	34.4	36.7
Barlow Twins (Zbontar et al., 2021)	75.7	47.2	50.3	82.6	56.8	63.4	59.0	39.2	42.5	56.0	34.3	36.5
MAE (He et al., 2022)	77.4	48.6	53.0	82.9	57.8	63.9	60.2	39.1	42.9	55.9	35.2	36.4
SwAV (Caron et al., 2020b)	75.5	46.5	49.6	82.6	56.1	62.7	58.6	38.4	41.3	55.2	33.8	35.9
MEC (Liu et al., 2022a)	77.4	48.3	52.3	82.8	57.5	64.5	59.8	39.8	43.2	56.3	34.7	36.8
RELIC v2 (Tomasev et al., 2022)	76.9	48.0	52.0	82.1	57.3	63.9	58.4	39.3	42.3	56.0	34.6	36.3
VICRegL (Bardes et al., 2022)	75.9	47.4	52.3	82.6	56.4	62.9	59.2	39.8	42.1	56.5	35.1	36.8
PIDSSL (Qiang et al., 2025b)	77.6	50.1	51.7	85.3	58.4	63.9	59.2	40.6	43.9	57.1	35.9	37.1
SimCLR + GeSSL	78.1	49.4	52.1	84.5	58.2	63.3	59.1	40.1	43.4	56.8	35.9	36.5
MoCo + GeSSL	78.7	50.0	**54.9**	**85.6**	60.5	65.9	61.5	**42.8**	**44.8**	**58.9**	37.0	39.0
BYOL + GeSSL	78.7	49.8	53.2	84.9	59.0	64.8	60.7	40.9	44.0	57.5	36.2	38.1
MAE + GeSSL	79.1	**51.0**	54.4	85.4	**61.2**	65.9	62.1	42.1	44.8	58.2	**38.3**	39.1
SimSiam + GeSSL	**79.3**	50.5	54.1	85.0	59.4	65.8	62.0	41.5	44.3	58.4	37.5	**39.6**
SwAV + GeSSL	78.4	49.3	52.3	84.8	58.7	65.1	61.3	40.7	43.9	57.0	36.6	38.8
VICRegL + GeSSL	78.9	50.5	54.6	85.4	59.8	**66.0**	**62.2**	42.2	44.5	58.7	37.8	39.4
PIDSSL + GeSSL	78.8	52.4	53.8	86.6	60.5	65.1	61.4	42.2	46.2	58.4	38.3	38.8

The bound separates three sources of error. The empirical term reflects how well GeSSL satisfies its observable training objectives; the complexity and finite-sample terms quantify generalization from observed to unseen self-supervised episodes; and $\varepsilon_{\text{proxy}}$ captures the unavoidable discrepancy between unlabeled SSL signals and downstream semantics. Accordingly, our analysis does not establish exact causal or semantic recovery. Rather, it shows that GeSSL optimizes an observable model-reliability objective that generalizes across self-supervised episodes and, when sufficiently aligned with downstream semantics, provides conditional control of the target risk.

5 Experiments

In this section, we conduct extensive experiments on various settings to verify the effectiveness of GeSSL. For unsupervised and semi-supervised learning, we select CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), STL-10 (Coates et al., 2011), Tiny ImageNet (Le & Yang, 2015), ImageNet-100 (Tian et al., 2020) and ImageNet (Deng et al., 2009a); For transfer learning, we select PASCAL VOC (Everingham et al., 2010), COCO (Lin et al., 2014a), Flower102 (Nilsback & Zisserman, 2008), Food101 (Bossard et al., 2014), etc.; For few-shot learning, we select Omniglot (Lake et al., 2019), miniImageNet (Vinyals et al., 2016a), CIFAR-FS (Bertinetto et al., 2018), CUB (Welinder et al., 2010), Cars (Krause et al., 2013), etc., for evaluation. We select both D-SSL and G-SSL baselines for comparison. All results are reported over five runs on NVIDIA 4090 GPUs. Implementation details, additional results, and further analyses, including relation reliability, ablation studies, nuisance robustness, and the correlation between the certification objective and downstream performance, are provided in Appendices B–E.

Table 4: Few-shot learning accuracies (± 95% confidence interval) on miniImageNet, Omniglot, and CIFAR-FS with C4. See Appendix C for full results.

Method	Omniglot			*mini*ImageNet			CIFAR-FS		
	(5,1)	(5,5)	(20,1)	(5,1)	(5,5)	(20,1)	(5,1)	(5,5)	(20,1)
Unsupervised Few-shot Learning									
CACTUs (Hsu et al., 2018)	65.29 ± 0.21	86.25 ± 0.19	49.54 ± 0.21	39.32 ± 0.28	53.54 ± 0.27	31.99 ± 0.29	40.02 ± 0.23	58.16 ± 0.22	35.88 ± 0.25
UMTRA (Khodadadeh et al., 2019)	83.32 ± 0.37	94.23 ± 0.35	75.84 ± 0.34	39.23 ± 0.34	51.78 ± 0.32	30.27 ± 0.34	41.61 ± 0.40	60.55 ± 0.38	37.10 ± 0.39
LASIUM (Khodadadeh et al., 2020)	82.38 ± 0.36	95.11 ± 0.36	70.23 ± 0.36	42.12 ± 0.38	54.98 ± 0.37	34.26 ± 0.35	45.33 ± 0.32	62.65 ± 0.33	38.40 ± 0.33
SVEBM (Kong et al., 2021)	87.07 ± 0.28	94.13 ± 0.27	73.33 ± 0.28	44.74 ± 0.29	58.38 ± 0.28	39.71 ± 0.30	47.24 ± 0.25	63.10 ± 0.28	40.10 ± 0.28
GMVAE (Lee et al., 2021)	90.89 ± 0.32	96.05 ± 0.32	81.51 ± 0.33	42.28 ± 0.36	56.97 ± 0.38	39.83 ± 0.36	47.45 ± 0.36	63.20 ± 0.35	41.55 ± 0.35
PsCo (Jang et al., 2023)	96.18 ± 0.21	98.22 ± 0.23	89.32 ± 0.23	46.35 ± 0.24	63.05 ± 0.23	40.84 ± 0.27	51.77 ± 0.27	69.12 ± 0.26	45.08 ± 0.27
Self-supervised Learning									
SimCLR (Chen et al., 2020a)	90.83 ± 0.21	97.67 ± 0.21	81.67 ± 0.23	42.32 ± 0.38	51.10 ± 0.37	36.36 ± 0.36	49.44 ± 0.30	60.02 ± 0.29	39.29 ± 0.30
MoCo (Chen et al., 2020b)	87.83 ± 0.20	95.52 ± 0.19	80.03 ± 0.21	40.56 ± 0.34	49.41 ± 0.37	36.52 ± 0.38	45.35 ± 0.31	58.11 ± 0.32	37.89 ± 0.32
MAE (He et al., 2022)	90.48 ± 0.21	97.45 ± 0.19	82.17 ± 0.11	41.45 ± 0.30	50.26 ± 0.20	35.18 ± 0.23	48.00 ± 0.24	60.51 ± 0.36	38.55 ± 0.31
SimCLR + GeSSL	94.35 ± 0.31	$\mathbf{98.41 \pm 0.19}$	90.23 ± 0.24	46.51 ± 0.29	62.56 ± 0.29	39.56 ± 0.12	52.72 ± 0.15	67.52 ± 0.11	46.81 ± 0.14
MoCo + GeSSL	93.15 ± 0.15	97.84 ± 0.14	88.84 ± 0.13	47.23 ± 0.14	61.05 ± 0.12	40.75 ± 0.11	51.58 ± 0.09	66.25 ± 0.08	44.48 ± 0.11
MAE + GeSSL	$\mathbf{95.99 \pm 0.09}$	98.00 ± 0.06	$\mathbf{92.58 \pm 0.17}$	$\mathbf{47.55 \pm 0.19}$	$\mathbf{63.97 \pm 0.13}$	$\mathbf{42.01 \pm 0.19}$	$\mathbf{52.80 \pm 0.16}$	$\mathbf{68.99 \pm 0.20}$	$\mathbf{47.11 \pm 0.23}$

5.1 Performance Comparison

Unsupervised Evaluation. Following the standard linear probing protocol (Chen et al., 2020a), we freeze the backbone and train a supervised linear classifier using Adam (Kingma & Ba, 2014) (momentum 0.8, weight decay 10^{-4}) for 500 epochs with batch size 128 and learning rate decayed from 5×10^{-2} to 5×10^{-6}. We employ ResNet-18 on small datasets (CIFAR-10/100, STL-10, Tiny ImageNet) and ResNet-50 on ImageNet-100 and ImageNet. Table 1 shows that GeSSL consistently outperforms competing SSL methods across all benchmarks (see Appendix C.1 for details).

Semi-supervised Learning. We adopt the commonly used protocol (Zbontar et al., 2021) and create two balanced subsets by sampling 1% and 10% of the training set. We fine-tune the models for 50 epochs with learning rates of 0.05 and 1.0 for the classifier, 0.0001 and 0.01 for the backbone on the 1% and 10% subsets. Table 2 shows that the performance with GeSSL is superior to the SOTA methods, e.g., when only 1% labels are available, the improvement reaches more than 3%.

Transfer Learning. We use Faster R-CNN (Ren et al., 2015) for VOC detection and Mask R-CNN (He et al., 2017) for COCO detection and segmentation with the same C4-backbone (Wu et al., 2019). We train the Faster R-CNN on the VOC 07+12 set (16K images) and reduce the initial learning rate by 10 at 18K and 22K iterations, while training on the VOC 07 set (5K images) with fewer iterations. For Mask R-CNN, we train it on the COCO 2017 train split and report on the val split. See Appendix C.2 for details. Table 3 shows the performance achieved by GeSSL. After introducing GeSSL, the models achieve SOTA performance.

Few-shot Learning. We adopt the commonly used protocol (Jang et al., 2023) on miniImageNet, Omniglot, and CIFAR-FS. For the few-shot SSL task, we randomly select N samples without class-level overlap for each task, and then apply 2-times data augmentation, obtaining a N-way 2-shot task with N classes and $2N$ samples. We use the SGD optimizer, setting the momentum and weight decay values to 0.9 and 10^{-4} respectively. We evaluate the trained model's performance in some unseen samples sampled from a new class. Table 4 shows the standard few-shot learning results of GeSSL compared with the baselines. Our framework still achieves remarkable performance improvement, demonstrating the superiority of GeSSL. See Appendix C.3 for more details.

5.2 Analysis and Ablation

Model efficiency. Figure 2 reports the trade-off between performance and efficiency on STL-10 (Coates et al., 2011). GeSSL introduces additional computation through the auxiliary relation boundary learner and the bi-level update, but the overhead remains moderate relative to the performance gains. More detailed memory and runtime analyses are provided in Appendix D.4.

Effect of the discriminative relation loss. The discriminative relation loss ($\mathcal{L}_{\text{disc}}$) serves as a proxy for counterfactual necessity, filtering out spurious relations driven solely by superficial similarity. To evaluate its impact, we visualize the learned feature representations before and after incorporating $\mathcal{L}_{\text{disc}}$ using t-SNE. As illustrated in Figure 8, adding $\mathcal{L}_{\text{disc}}$ yields visibly sharper cluster boundaries across multiple SSL baselines. This confirms that the adaptive relation boundary enhances discriminability by suppressing unreliable relation assignments.

Effect of support-query bi-level optimization. Our bi-level setup enforces counterfactual sufficiency: relations learned on one perturbed set must remain valid on another. We compare this

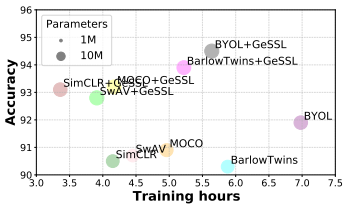


Figure 2: Model efficiency.

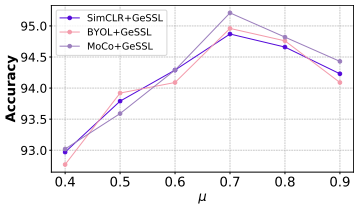


Figure 3: Ablation study of μ.

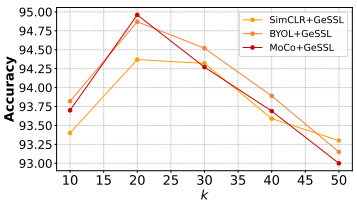


Figure 4: Ablation study of k.

against (i) joint single-stage optimization and (ii) learning a distinct adapted model f' per mini-batch. Figure 11 shows clear gains from bi-level optimization, indicating that explicitly validating support updates on query relations is superior to simply aggregating objectives.

Parameter sensitivity. We evaluate the sensitivity of (i) μ: controls the strength of the discriminative relation loss, and (ii) k: controls how sharply the adaptive boundary separates compatible and incompatible relations. Figures 3-4 show that GeSSL is reasonably stable around the selected settings, with the best performance obtained at $\mu = 0.7$ and $k = 30$. See Appendix D.3 for more results.

6 Related Work

Self-supervised representation learning. SSL learns representations from unlabeled data by constructing pretext signals. Existing methods are commonly grouped into discriminative and generative paradigms (Jaiswal et al., 2020; Kang et al., 2023). Discriminative methods, such as SimCLR (Chen et al., 2020a), BYOL (Grill et al., 2020), and Barlow Twins (Zbontar et al., 2021), enforce consistency between augmented views, while generative methods, such as MAE (He et al., 2022) and VideoMAE (Tong et al., 2022), recover missing or corrupted content. These methods have achieved strong empirical performance, but they usually optimize constructed relations directly. Our work instead studies whether these self-supervised relations are reliable.

Relations and pseudo-supervision in SSL. Many SSL objectives can be viewed as learning from pseudo-supervision induced by augmentations, instance identities, clustering assignments, nearest neighbors, or reconstruction targets (Caron et al., 2018; 2020a; Dwibedi et al., 2021). However, such relations may be noisy: different instances can share semantics and become false negatives, while same-instance views may share nuisance factors rather than semantic content (Chuang et al., 2020; Huynh et al., 2022). Prior work mainly improves relation construction or negative sampling. In contrast, we ask when a constructed relation should be trusted, and formulate SSL as counterfactual relation certification.

Causal and invariant representation learning. Causal and invariant learning aim to capture stable mechanisms rather than environment-specific correlations (Arjovsky et al., 2019; Krueger et al., 2021; Schölkopf et al., 2021; Ahuja et al., 2023). Recent SSL studies also analyze confounding, spurious correlations, and OOD generalization from causal perspectives (Li et al., 2022; Qiang et al., 2025a;b). Our work is related but different: we do not claim to recover true causal variables or counterfactual samples. Instead, we use necessity and sufficiency as relation-level principles, and approximate them through observable SSL objectives.

7 Conclusion

In this paper, we identify a fundamental reliability issue in SSL: models may satisfy self-supervised objectives by exploiting spurious dependencies rather than semantically meaningful factors. To address this issue, we introduce a counterfactual perspective that characterizes reliable SSL models through two complementary principles: sensitivity to semantic changes and stability under nuisance variations. Based on these principles, we propose GeSSL, which approximates counterfactual necessity with a discriminative relation loss and counterfactual sufficiency with consistency at the sample, set, and distribution levels. Theoretical analysis establishes bounded generalization error on unseen tasks, while extensive experiments show consistent improvements over representative discriminative and generative SSL baselines with moderate computational overhead.

AI USAGE STATEMENT

In this work, we used generative AI tools for grammar and language polishing. They were not involved in research ideation, methodology, analysis, or substantive writing. The authors take full responsibility for all contents of the paper.

REFERENCES

Aviad Aberdam, Ron Litman, Shahar Tsiper, Oron Anschel, Ron Slossberg, Shai Mazor, R Manmatha, and Pietro Perona. Sequence-to-sequence contrastive learning for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15302–15312, 2021.

Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. Interventional causal representation learning. In *International conference on machine learning*, pp. 372–407. PMLR, 2023.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *International Conference on Machine Learning*, pp. 1416–1429. PMLR, 2023.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicregl: Self-supervised learning of local visual features. *arXiv preprint arXiv:2210.01571*, 2022.

Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pp. 446–461. Springer, 2014.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision*, pp. 132–149, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924, 2020a.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020b.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Shuo Chen, Gang Niu, Chen Gong, Jun Li, Jian Yang, and Masashi Sugiyama. Large-margin contrastive learning with distance polarization regularizer. In *International Conference on Machine Learning*, pp. 1673–1683. PMLR, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Sang Keun Choe, Willie Neiswanger, Pengtao Xie, and Eric Xing. Betty: An automatic differentiation library for multilevel optimization. *arXiv preprint arXiv:2207.02849*, 2022.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8765–8775, 2020.

Adam Coates, Andrew Ng, and Honglak Lee. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, June 2011.

Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pp. 168–172. IEEE, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009a. doi: 10.1109/CVPR.2009.5206848.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009b.

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

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9588–9597, 2021.

Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62, 2022.

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pp. 3015–3024. PMLR, 2021.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017a.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017b.

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pp. 3748–3758. PMLR, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pp. 124–141. Springer, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning. *arXiv preprint arXiv:1810.02334*, 2018.

Weiran Huang, Mingyang Yi, and Xuyang Zhao. Towards the generalization of contrastive self-supervised learning. *arXiv preprint arXiv:2111.00743*, 2021.

Tri Huynh, Simon Kornblith, Matthew R. Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2785–2795, 2022.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Huiwon Jang, Hankook Lee, and Jinwoo Shin. Unsupervised meta-learning via few-shot pseudo-supervised contrastive learning. *arXiv preprint arXiv:2303.00996*, 2023.

Siddharth Joshi and Baharan Mirzasoleiman. Data-efficient contrastive self-supervised learning: Most beneficial examples for supervised learning contribute the least. In *International conference on machine learning*, pp. 15356–15370. PMLR, 2023.

Mingu Kang, Heon Song, Seonwook Park, Donggeun Yoo, and Sérgio Pereira. Benchmarking self-supervised learning on diverse pathology datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3344–3354, 2023.

Siavash Khodadadeh, Ladislau Boloni, and Mubarak Shah. Unsupervised meta-learning for few-shot image classification. *Advances in neural information processing systems*, 32, 2019.

Siavash Khodadadeh, Sharare Zehtabian, Saeed Vahidian, Weijia Wang, Bill Lin, and Ladislau Bölöni. Unsupervised meta-learning through latent-space interpolation in generative models. *arXiv preprint arXiv:2006.10236*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Deqian Kong, Bo Pang, and Ying Nian Wu. Unsupervised meta-learning via latent space energy-based model of symbol vector coupling. In *Fifth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*, 2021.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Rémi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation. In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. The omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, 29:97–104, 2019.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Dong Bok Lee, Dongchan Min, Seanie Lee, and Sung Ju Hwang. Meta-gmvae: Mixture of gaussian vae for unsupervised meta-learning. In *International Conference on Learning Representations*, 2021.

Jiangmeng Li, Wenwen Qiang, Yanan Zhang, Wenyi Mo, Changwen Zheng, Bing Su, and Hui Xiong. Metamask: Revisiting dimensional confounder for self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38501–38515, 2022.

Yazhe Li, Roman Pogodin, Danica J Sutherland, and Arthur Gretton. Self-supervised learning with kernel dependence maximization. *Advances in Neural Information Processing Systems*, 34: 15543–15556, 2021.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014a.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014b.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Xin Liu, Zhongdao Wang, Ya-Li Li, and Shengjin Wang. Self-supervised learning via maximum entropy coding. *Advances in Neural Information Processing Systems*, 35:34091–34105, 2022a.

Xin Liu, Zhongdao Wang, Ya-Li Li, and Shengjin Wang. Self-supervised learning via maximum entropy coding. *Advances in Neural Information Processing Systems*, 35:34091–34105, 2022b.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International conference on artificial intelligence and statistics*, pp. 1540–1552. PMLR, 2020.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

Abdelrahman Mohamed, Dmitry Demidov, and Zakaria Sebaitre. Image captioning through self-supervised learning. Technical report, Technical Report, 2022.

Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Serdar Ozsoy, Shadi Hamdan, Sercan Arik, Deniz Yuret, and Alper Erdogan. Self-supervised learning with an information maximization criterion. *Advances in Neural Information Processing Systems*, 35:35240–35253, 2022.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2 edition, 2009.

Judea Pearl, Madelyn Glymour, and Nicholas P. Jewell. *Causal Inference in Statistics: A Primer*. John Wiley & Sons, Chichester, UK, 2016.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–732, 2016.

Wenwen Qiang, Zeen Song, Ziyin Gu, Jiangmeng Li, Changwen Zheng, Fuchun Sun, and Hui Xiong. On the generalization and causal explanation in self-supervised learning. *International Journal of Computer Vision*, 133(4):1727–1754, 2025a. doi: 10.1007/s11263-024-02263-9.

Wenwen Qiang, Jingyao Wang, Zeen Song, Jiangmeng Li, and Changwen Zheng. On the out-of-distribution generalization of self-supervised learning. In *International Conference on Machine Learning*, 2025b.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850. PMLR, 2016.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33: 8728–8740, 2020.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1285–1293, 2012.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Multiview Coding, December 2020.

Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? *arXiv preprint arXiv:2201.05119*, 2022.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016a.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2016b.

Jingyao Wang, Luntian Mou, Changwen Zheng, and Wen Gao. Image-based freeform handwriting authentication with energy-oriented self-supervised learning. *IEEE Transactions on Multimedia*, 27:1397–1409, 2024a.

Jingyao Wang, Wenwen Qiang, Xingzhe Su, Changwen Zheng, Fuchun Sun, and Hui Xiong. Towards task sampler learning for meta-learning. *International Journal of Computer Vision*, pp. 1–31, 2024b.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

Zhongdao Wang, Hengshuang Zhao, Ya-Li Li, Shengjin Wang, Philip Torr, and Luca Bertinetto. Do different tracking tasks require different appearance models? *Advances in Neural Information Processing Systems*, 34:726–738, 2021.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

Linshan Wu, Jiaxin Zhuang, and Hao Chen. Voco: A simple-yet-effective volume contrastive learning framework for 3d medical image analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22873–22882, 2024.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Ujala Yasmeen, Jamal Hussain Shah, Muhammad Attique Khan, Ghulam Jillani Ansari, Saeed Ur Rehman, Muhammad Sharif, Seifedine Kadry, and Yunyoung Nam. Text detection and classification from low quality natural images. *Intell. Autom. Soft Comput*, 26(4):1251–1266, 2020.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. *Advances in neural information processing systems*, 32, 2019.

Xueting Zhang, Debin Meng, Henry Gouk, and Timothy M Hospedales. Shallow bayesian meta learning for real-world few-shot recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 651–660, 2021.

Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Ressl: Relational self-supervised learning with weak augmentation. *Advances in Neural Information Processing Systems*, 34:2543–2555, 2021.

Yu Zheng. Methodologies for cross-domain data fusion: An overview. *IEEE transactions on big data*, 1(1):16–34, 2015.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

APPENDIX

The appendix is organized into several sections:

- Appendix A contains the analyses and proofs of the presented definitions and theorems.
- Appendix B presents the implementation and architecture of our GeSSL.
- Appendix C showcases additional experiments, full results, and experimental details of the comparison experiments that were omitted in the main text due to space limitations.
- Appendix D provides the additional experiments and full details of the ablation studies that were omitted in the main text due to page limitations.
- Appendix E provides the additional experiments for the counterfactual relation certification that were omitted in the main text due to page limitations.
- Appendix F provides the discussion about the proposed methodology.

Note that before we illustrate the details and analysis, we provide a brief summary about all the experiments conducted in this paper, as shown in Table 5.

Table 5: Illustration of the experiments conducted in this work. Note that all experimental results are obtained after five rounds of experiments.

Experiments	**Location**	**Results**
Experiments of unsupervised learning on six benchmark dataset	Section 5.1 and Appendix C.1	Table 1, Table 6, Table 7, and Table 11
Experiments of semi-supervised learning on on ImageNet with two settings	Section 5.1	Table 2 and Table 12
Experiment of transfer learning	Section 5.1 and Appendix C.2	Table 3, Table 8, and Table 9
Experiment of few-shot learning on standard and cross-domain scenarios	Section 5.1 and Appendix C.3	Table 4 and Table 10
Ablation study-Model efficiency	Section 5.2 and Appendix D.1	Figure 2 and Table 19
Ablation study-Effect of $\mathcal{L}_{disc}$	Section 5.2 and Appendix D.2	Figure 8
Visualization Analyses	Appendix C.7	Figure 7
Ablation study-Parameter sensitivity	Section 5.2 and Appendix D.3	Figure 3, Figure 9, and Figure 10
Ablation study-Evaluation of the bi-level optimization	Section 5.2 and Appendix D.4	Figure 12
Evaluation of Distribution-level Consistency	Appendix C.4	Figure 5 and Table 13
Evaluation of generative SSL on three scenarios	Appendix C.5	Figure 6, Table 14, Table 15, and Table 16
Experiments for the counterfactual relation certification	Appendix E	Table 23, Table 24, Table 25, and Table 26
Evaluation on more modalities	Appendix C.6	Table 17

A Proofs for the Theoretical Analysis

This appendix provides the proof of Theorem 4.1 and Corollary 4.3. We emphasize that the analysis treats each support-query pair $\tau = (S_\tau, Q_\tau)$ as one self-supervised episode. Therefore, independence between S_τ and Q_τ within the same episode is not required. The only sampling assumption used in the generalization result is that the episodes $\{\tau_i\}_{i=1}^M$ are sampled independently from the episode distribution $\mathcal{T}$.

A.1 Rademacher Complexity

Recall the GeSSL function class

$$\mathcal{G} = \{\tau \mapsto \mathcal{J}(\theta, \phi; \tau) : (\theta, \phi) \in \Theta \times \Phi\}. \tag{10}$$

For M episodes $\tau_{1:M} = \{\tau_i\}_{i=1}^M$, the empirical Rademacher complexity is defined as

$$\widehat{\mathfrak{R}}_M(\mathcal{G}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}} \frac{1}{M}\sum_{i=1}^M \sigma_i g(\tau_i)\right], \tag{11}$$

where $\sigma_1, \ldots, \sigma_M$ are independent Rademacher variables taking values in $\{-1, +1\}$ with equal probability.

The expected Rademacher complexity is

$$\mathfrak{R}_M(\mathcal{G}) = \mathbb{E}_{\tau_{1:M}\sim\mathcal{T}^M}\left[\widehat{\mathfrak{R}}_M(\mathcal{G})\right]. \tag{12}$$

The function $g_{\theta,\phi}(\tau) = \mathcal{J}(\theta, \phi; \tau)$ already includes the support-dependent adaptation $A(\theta, \phi, S_\tau)$. Hence, the complexity of the inner-loop update is implicitly contained in $\mathcal{G}$. No independence between the support and query sets is required for the uniform-convergence argument.

A.2 Proof of Theorem 4.1

For any $g \in \mathcal{G}$, define its population and empirical risks as

$$R(g) = \mathbb{E}_{\tau\sim\mathcal{T}}[g(\tau)] \tag{13}$$

and

$$\widehat{R}_M(g) = \frac{1}{M}\sum_{i=1}^M g(\tau_i), \tag{14}$$

respectively.

We assume that $0 \le g(\tau) \le B$ for all $g \in \mathcal{G}$ and all τ. By the standard Rademacher uniform-convergence bound, with probability at least $1 - \delta$,

$$\sup_{g\in\mathcal{G}}\left[R(g) - \widehat{R}_M(g)\right] \le 2\mathfrak{R}_M(\mathcal{G}) + B\sqrt{\frac{\log(1/\delta)}{2M}}. \tag{15}$$

For completeness, Eq. equation 15 follows from the usual symmetrization argument:

$$\mathbb{E}\left[\sup_{g\in\mathcal{G}}\left(R(g) - \widehat{R}_M(g)\right)\right] \le 2\mathfrak{R}_M(\mathcal{G}),$$

together with bounded-difference concentration for functions taking values in $[0, B]$.

Substituting $g_{\theta,\phi}(\tau) = \mathcal{J}(\theta, \phi; \tau)$ into Eq. equation 15 gives, simultaneously for every (θ, ϕ),

$$\mathbb{E}_{\tau\sim\mathcal{T}}\left[\mathcal{J}(\theta, \phi; \tau)\right] \le \frac{1}{M}\sum_{i=1}^M \mathcal{J}(\theta, \phi; \tau_i) + 2\mathfrak{R}_M(\mathcal{G}) + B\sqrt{\frac{\log(1/\delta)}{2M}}.$$

Because the inequality holds uniformly over the entire function class, it remains valid after choosing $(\hat{\theta}, \hat{\phi})$ as an arbitrary data-dependent output of the GeSSL training procedure. Therefore,

$$\mathbb{E}_{\tau\sim\mathcal{T}}\left[\mathcal{J}(\hat{\theta}, \hat{\phi}; \tau)\right] \le \frac{1}{M}\sum_{i=1}^M \mathcal{J}(\hat{\theta}, \hat{\phi}; \tau_i) + 2\mathfrak{R}_M(\mathcal{G}) + B\sqrt{\frac{\log(1/\delta)}{2M}}.$$

This proves Theorem 4.1. □

A.3 Proof of Corollary 4.3

By Assumption 4.2,

$$\mathbb{E}_{\tau\sim\mathcal{T}}\left[\mathcal{L}_{tar}\big(A(\hat{\theta},\hat{\phi},S_\tau);\tau\big)\right] \le \alpha\mathbb{E}_{\tau\sim\mathcal{T}}\left[\mathcal{J}(\hat{\theta},\hat{\phi};\tau)\right] + \varepsilon_{\text{proxy}}.$$

Applying Theorem 4.1 to the population GeSSL objective on the right-hand side yields

$$\mathbb{E}_{\tau\sim\mathcal{T}}\left[\mathcal{L}_{tar}\big(A(\hat{\theta},\hat{\phi},S_\tau);\tau\big)\right] \le \alpha\left[\frac{1}{M}\sum_{i=1}^{M}\mathcal{J}(\hat{\theta},\hat{\phi};\tau_i) + 2\mathfrak{R}_M(\mathcal{G}) + B\sqrt{\frac{\log(1/\delta)}{2M}}\right] + \varepsilon_{\text{proxy}},$$

which is exactly Eq. equation 9. □

A.4 Discussion of the Assumptions

The boundedness assumption is used only to obtain a standard high-probability uniform-convergence bound. In practice, bounded representations, normalized similarities, bounded parameter domains, or clipped analytical losses are sufficient to satisfy this condition. More general sub-Gaussian assumptions can be used instead, but they do not change the main interpretation of the result.

The Rademacher term $\mathfrak{R}_M(\mathcal{G})$ captures the effective complexity of the complete GeSSL learning procedure, including the encoder, the adaptive relation-boundary learner, and the support-dependent inner update. The bound is informative when this complexity grows slower than the number of training episodes.

Assumption 4.2 is qualitatively different from the boundedness condition. It is a task-dependent alignment assumption connecting the observable self-supervised objective with an unobserved downstream target. Without such an assumption, no unlabeled SSL objective can in general guarantee performance on an arbitrary downstream labeling function. The term $\varepsilon_{\text{proxy}}$ therefore makes explicit the semantic mismatch that cannot be eliminated by self-supervised training alone.

Finally, the theoretical results should not be interpreted as proving counterfactual identification. Counterfactual necessity and sufficiency motivate the design of the observable GeSSL objectives, while the analysis above establishes generalization properties of these proxies. The causal terminology characterizes the desired behavior of the learned representation; it does not imply recovery of a latent causal model from unlabeled observations.

B Implementation Details

We use C4-backbone, ResNet-18, and ResNet-50 backbones as our encoders for a fair comparison with different methods. The convolutional layers are followed by batch normalization, ReLU nonlinearity, and max pooling (strided convolution), respectively. The last layer is fed into a MLP for $\mathcal{L}_{disc}$. These architectures are pre-trained and kept fixed during training. We optimize our model with a Stochastic Gradient Descent (SGD) optimizer, setting the momentum and weight decay values to 0.9 and 10^{-4} respectively. The specific adjustments of the experimental settings corresponding to different experiments are illustrated in Section 5.1 of the main text. All the experiments are apples-to-apples comparisons and performed on NVIDIA RTX 4090 GPUs. We build tasks based on images with a batch size of $B = 16$. For data augmentation, we use the same data augmentation scheme as SimCLR to augment each image in the batch 5 times. In simple terms, we draw a random patch (224×224) from the original image, and then apply a random augmentation sequence composed of random horizontal flip, cropping, color jitter, etc.

C Additional Experiments

C.1 Unsupervised Learning

In this section, we present additional results of the unsupervised learning experiments. Specifically, Table 6 shows the results on four small-scale datasets. We can observe that applying the proposed GeSSL framework significantly outperforms the state-of-the-art (SOTA) methods on all four datasets.

Table 6: The classification accuracies ($\pm$ 95% confidence interval) of a linear classifier (linear) and a 5-nearest neighbors classifier (5-nn) with a ResNet-18 as the feature extractor. The comparison baselines cover almost all types of methods mentioned in Section 6. The "-" denotes that the results are not reported.

Method	CIFAR-10		CIFAR-100		STL-10		Tiny ImageNet	
	linear	**5 − nn**	**linear**	**5 − nn**	**linear**	**5 − nn**	**linear**	**5 − nn**
SimCLR (Chen et al., 2020a)	91.80 ± 0.15	88.42 ± 0.15	66.83 ± 0.27	56.56 ± 0.18	90.51 ± 0.14	85.68 ± 0.10	48.84 ± 0.15	32.86 ± 0.25
MoCo (Chen et al., 2020b)	91.69 ± 0.12	88.66 ± 0.14	67.02 ± 0.16	56.29 ± 0.25	90.64 ± 0.28	88.01 ± 0.19	50.92 ± 0.22	35.55 ± 0.16
BYOL (Grill et al., 2020)	91.93 ± 0.22	89.45 ± 0.22	66.60 ± 0.16	56.82 ± 0.17	91.99 ± 0.13	88.64 ± 0.20	51.00 ± 0.12	36.24 ± 0.28
SimSiam (Chen & He, 2021)	91.71 ± 0.27	88.65 ± 0.17	67.22 ± 0.26	56.36 ± 0.19	91.01 ± 0.19	88.16 ± 0.19	51.14 ± 0.20	35.67 ± 0.16
Barlow Twins (Zbontar et al., 2021)	90.88 ± 0.19	89.68 ± 0.21	66.13 ± 0.10	56.70 ± 0.25	90.38 ± 0.13	87.13 ± 0.23	49.78 ± 0.26	34.18 ± 0.18
SwAV (Caron et al., 2020b)	91.03 ± 0.19	89.52 ± 0.24	66.56 ± 0.17	57.01 ± 0.25	90.72 ± 0.29	86.24 ± 0.26	52.02 ± 0.26	37.40 ± 0.11
DINO (Caron et al., 2021)	91.83 ± 0.25	90.15 ± 0.33	67.15 ± 0.21	56.48 ± 0.19	91.03 ± 0.12	86.15 ± 0.25	51.13 ± 0.30	37.86 ± 0.19
W-MSE (Ermolov et al., 2021)	91.99 ± 0.12	89.87 ± 0.25	67.64 ± 0.16	56.45 ± 0.26	91.75 ± 0.23	88.59 ± 0.15	49.22 ± 0.16	35.44 ± 0.10
RELIC v2 (Tomasev et al., 2022)	91.92 ± 0.14	90.02 ± 0.22	67.66 ± 0.20	57.03 ± 0.18	91.10 ± 0.23	88.66 ± 0.12	49.33 ± 0.13	35.52 ± 0.22
LMCL (Chen et al., 2021)	91.91 ± 0.25	88.52 ± 0.29	67.01 ± 0.18	56.86 ± 0.14	90.87 ± 0.18	85.91 ± 0.25	49.24 ± 0.18	32.88 ± 0.13
ReSSL (Zheng et al., 2021)	90.20 ± 0.16	88.26 ± 0.18	66.79 ± 0.12	53.72 ± 0.28	88.25 ± 0.14	86.33 ± 0.17	46.60 ± 0.18	32.39 ± 0.20
SSL-HSIC (Li et al., 2021)	91.95 ± 0.14	89.99 ± 0.17	67.23 ± 0.26	57.01 ± 0.27	92.09 ± 0.20	88.91 ± 0.29	51.37 ± 0.15	36.03 ± 0.12
CorInfoMax (Ozsoy et al., 2022)	91.81 ± 0.11	89.85 ± 0.13	67.09 ± 0.24	56.92 ± 0.23	91.85 ± 0.25	89.99 ± 0.24	51.23 ± 0.14	35.98 ± 0.09
MEC (Liu et al., 2022a)	90.55 ± 0.22	87.80 ± 0.10	67.36 ± 0.27	57.25 ± 0.25	91.33 ± 0.14	89.03 ± 0.33	50.93 ± 0.13	36.28 ± 0.14
VICRegL (Bardes et al., 2022)	90.99 ± 0.13	88.75 ± 0.26	68.03 ± 0.32	57.34 ± 0.29	92.12 ± 0.26	90.01 ± 0.20	51.52 ± 0.13	36.24 ± 0.16
VoCo (Wu et al., 2024)	91.19 ± 0.10	89.12 ± 0.31	68.69 ± 0.39	57.81 ± 0.20	92.46 ± 0.21	90.54 ± 0.20	52.37 ± 0.15	37.12 ± 0.15
SimCLR + GeSSL	93.45 ± 0.21	91.35 ± 0.14	69.72 ± 0.15	58.80 ± 0.16	93.45 ± 0.22	91.72 ± 0.14	53.92 ± 0.17	37.49 ± 0.21
MoCo + GeSSL	93.05 ± 0.18	89.48 ± 0.20	68.48 ± 0.12	59.44 ± 0.18	93.42 ± 0.15	89.16 ± 0.26	52.34 ± 0.13	37.35 ± 0.11
BYOL + GeSSL	**94.05 ± 0.19**	**92.60 ± 0.28**	69.45 ± 0.18	59.15 ± 0.14	94.55 ± 0.16	90.73 ± 0.15	**55.12 ± 0.16**	37.76 ± 0.22
Barlow Twins + GeSSL	93.18 ± 0.16	91.23 ± 0.14	69.85 ± 0.16	60.12 ± 0.14	93.98 ± 0.08	89.76 ± 0.23	52.85 ± 0.12	35.39 ± 0.14
SwAV + GeSSL	93.37 ± 0.19	90.24 ± 0.22	70.28 ± 0.19	59.63 ± 0.20	93.05 ± 0.26	**91.92 ± 0.21**	52.12 ± 0.22	37.05 ± 0.30
DINO + GeSSL	93.08 ± 0.21	92.38 ± 0.22	**71.15 ± 0.16**	**62.03 ± 0.31**	**94.65 ± 0.24**	91.67 ± 0.18	53.74 ± 0.22	38.12 ± 0.21
VoCo + GeSSL	93.10 ± 0.01	91.16 ± 0.37	70.53 ± 0.30	59.34 ± 0.19	94.10 ± 0.22	92.01 ± 0.26	54.37 ± 0.06	**38.76 ± 0.10**

Table 6 shows the results on four small-scale datasets. The results still demonstrate the proposed GeSSL's ability to enhance the performance of self-supervised learning methods, achieving significant improvements over the original models on all baselines. Moreover, applying our GeSSL framework to all four types of representative SSL models as described in Section 6, including SimCLR, MoCo, BYOL, Barlow Twins, SwAV, and DINO, achieves an average improvement of 3% compared to the original frameworks. Table 7 provides the comparison results of our proposed GeSSL on a large-scale dataset, i.e., ImageNet. The results show that, (i) the self-supervised learning model applying GeSSL achieves the state-of-the-art result (SOTA) performance under all epoch conditions; and (ii) after applying the proposed GeSSL, the self-supervised learning models consistently outperforms the original frameworks in terms of average classification accuracy at 100, 200 and 400 epochs. For 1000 epochs, VICRegL + GeSSL yields the best result among other state-of-the-art methods, with an average accuracy of 78.72%.

More recent methods. The effect of GeSSL is reflected in the performance improvement when applying it to the SSL baselines. The experimental results above have demonstrated that after the introduction of GeSSL, the effects of all SSL baselines have been significantly improved. These results have shown the outstanding effectiveness and robustness of GeSSL. The SSL baselines we use cover all SOTA methods on the leaderboard of the adopted benchmark datasets (before submission). The methods proposed in 2025-26 are mainly variants of the currently used comparison baselines.

To evaluate the effect of GeSSL on recently proposed methods, we select the recently proposed SSL methods for testing (Baevski et al., 2023; Joshi & Mirzasoleiman, 2023; Wang et al., 2024a), where we follow the same experimental settings. The results are shown in Tables 11 and 12. The results still prove the effectiveness of GeSSL. We will supplement these results in the final version.

C.2 Transfer Learning

As mentioned in Section 5.1, we construct three sets of transfer learning experiments, including the most commonly used object detection and instance segmentation protocol (Chen et al., 2020a; Zbontar et al., 2021; Grill et al., 2020), transfer to other domains (different datasets), and transfer learning on video-based tasks. The results of the first experiment are illustrated in Section 5.1, and the other two sets of experiments are described below.

Transfer to other domains. To explore the nature of transfer learning of the proposed GeSSL, we leverage models that had been pre-trained on the CIFAR100 dataset, including SimCLR (Chen et al., 2020a), BYOL (Grill et al., 2020), and Barlow Twins (Zbontar et al., 2021), on the CIFAR100 dataset. We then applied these models to four distinct datasets, including CIFAR10 (Krizhevsky

Table 7: The Top-1 and Top-5 classification accuracies of linear classification on the ImageNet dataset with ResNet-50 as the feature extractor. We record the comparison results from 100, 200, 400, and 1000 epochs.

Method	100 Epochs		200 Epochs		400 Epochs	1000 Epochs
	Top-1	**Top-5**	**Top-1**	**Top-5**	**Top-1**	**Top-1**
Supervised	71.93	-	73.45	-	74.92	76.35
SimCLR (Chen et al., 2020a)	66.54 ± 0.22	88.14 ± 0.26	68.32 ± 0.31	89.76 ± 0.23	69.24 ± 0.21	70.45 ± 0.30
MoCo (Chen et al., 2020b)	64.53 ± 0.25	86.17 ± 0.11	67.55 ± 0.27	88.42 ± 0.11	69.76 ± 0.14	71.16 ± 0.23
BYOL (Grill et al., 2020)	67.65 ± 0.27	88.95 ± 0.11	69.94 ± 0.21	89.45 ± 0.27	71.85 ± 0.12	73.35 ± 0.27
SimSiam (Chen & He, 2021)	68.14 ± 0.26	87.12 ± 0.26	70.02 ± 0.14	88.76 ± 0.23	70.86 ± 0.34	71.37 ± 0.22
Barlow Twins (Zbontar et al., 2021)	67.24 ± 0.22	88.66 ± 0.19	69.94 ± 0.32	88.97 ± 0.27	70.22 ± 0.15	73.29 ± 0.13
SwAV (Caron et al., 2020b)	66.55 ± 0.27	88.42 ± 0.22	69.12 ± 0.24	89.38 ± 0.20	70.78 ± 0.34	75.32 ± 0.11
DINO (Caron et al., 2021)	67.23 ± 0.19	88.48 ± 0.21	70.58 ± 0.24	91.32 ± 0.27	71.98 ± 0.26	73.94 ± 0.29
W-MSE (Ermolov et al., 2021)	67.48 ± 0.29	90.39 ± 0.27	70.85 ± 0.31	91.57 ± 0.20	72.49 ± 0.24	72.84 ± 0.18
RELIC v2 (Tomasev et al., 2022)	66.38 ± 0.23	90.89 ± 0.21	70.98 ± 0.21	91.15 ± 0.26	71.84 ± 0.21	72.17 ± 0.20
LMCL (Chen et al., 2021)	66.75 ± 0.13	89.85 ± 0.36	70.83 ± 0.26	90.04 ± 0.21	72.53 ± 0.24	72.97 ± 0.29
ReSSL (Zheng et al., 2021)	67.41 ± 0.27	90.55 ± 0.23	69.92 ± 0.24	91.25 ± 0.12	72.46 ± 0.29	72.91 ± 0.30
CorInfoMax (Ozsoy et al., 2022)	70.13 ± 0.12	91.14 ± 0.25	70.83 ± 0.15	91.53 ± 0.22	73.28 ± 0.24	74.87 ± 0.36
MEC (Liu et al., 2022a)	69.91 ± 0.10	90.67 ± 0.15	70.34 ± 0.27	91.25 ± 0.38	72.91 ± 0.27	75.07 ± 0.24
VICRegL (Bardes et al., 2022)	69.99 ± 0.25	91.27 ± 0.16	70.24 ± 0.27	91.60 ± 0.24	72.14 ± 0.20	75.07 ± 0.23
SimCLR + GeSSL	68.45 ± 0.20	89.62 ± 0.23	69.88 ± 0.21	91.32 ± 0.25	71.50 ± 0.16	72.82 ± 0.28
MoCo + GeSSL	66.78 ± 0.19	88.41 ± 0.20	69.60 ± 0.30	91.28 ± 0.39	70.82 ± 0.29	73.04 ± 0.22
SimSiam + GeSSL	70.61 ± 0.18	88.61 ± 0.17	72.04 ± 0.22	89.43 ± 0.40	72.78 ± 0.17	74.78 ± 0.24
Barlow Twins + GeSSL	69.62 ± 0.21	89.55 ± 0.19	72.84 ± 0.26	89.50 ± 0.19	74.10 ± 0.18	75.02 ± 0.22
SwAV + GeSSL	69.05 ± 0.18	89.50 ± 0.17	72.28 ± 0.19	90.68 ± 0.30	72.88 ± 0.18	76.38 ± 0.19
DINO + GeSSL	69.55 ± 0.20	90.62 ± 0.22	73.52 ± 0.30	**94.05 ± 0.26**	74.02 ± 0.26	76.40 ± 0.21
VICRegL + GeSSL	**72.75 ± 0.21**	**91.45 ± 0.18**	**73.91 ± 0.36**	93.77 ± 0.35	**74.38 ± 0.23**	**78.85 ± 0.29**

Table 8: The performance of adding task information in self-supervised models on different datasets.

Evl.dataset	SimCLR+GeSSL	BYOL+GeSSL	Barlow Twins+GeSSL	VICRegL+GeSSL
CIFAR10	+3.56	+2.51	+2.17	+2.80
Flower102	+4.03	+2.09	+2.94	+3.07
Food101	+1.85	+2.31	+2.01	+2.02
Aircraft	+2.57	+2.89	+2.24	+2.34

et al., 2009), Flower102 (Nilsback & Zisserman, 2008), Food101 (Bossard et al., 2014), and Aircraft (Maji et al., 2013). We first calculate the classification performance (Top-1) based on the existing self-supervised model on different data sets, recorded as $acc(\text{method}, \text{dataset})$, such as $acc(\text{SimCLR}, \text{Flower102})$. Then, we calculate the model's classification performance by incorporating GeSSL on those data sets, which is recorded as $acc(\text{method} + \text{GeSSL}, \text{dataset})$. Finally, we get the improvement $\Delta(\text{method}, \text{dataset}) = acc(\text{method} + \text{GeSSL}, \text{dataset}) - acc(\text{method}, \text{dataset})$ in classification performance on each dataset, as shown in Table 8. The results show that the migration effect of the model after applying GeSSL has been steadily improved, proving that GeSSL representations successfully capture invariant semantic factors, enhancing transferability across domains.

Video-based Task. In order to assess the performance of our method with video-based tasks, we transition our pre-trained model to handle a variety of video tasks, utilizing the UniTrack evaluation framework (Wang et al., 2021) as our testing ground. The findings are compiled in Table 9, which includes results from five distinct tasks, drawing on the features from [layer3/layer4] of the Resnet-50. The data indicates that existing SSL methods incorporating our GeSSL significantly surpass original SSL approaches, with SimCLR achieving more than a 2% improvement in VOS (Perazzi et al., 2016), and BYOL seeing over a 3% gain in MOT (Milan et al., 2016).

C.3 Few-shot Learning

The outstanding performance of GeSSL in the few-shot learning scenario has been confirmed in Section 5.1, where it can produce good results with limited data. However, the situation becomes complicated in scenarios where data collection is infeasible in real life, such as medical diagnosis and satellite imagery (Zheng, 2015; Tang et al., 2012). Therefore, the performance of the model on cross-domain few-shot learning tasks is crucial, as it determines the applicability of the learning

Table 9: Transfer learning on video tracking tasks. All methods use the same ResNet-50 backbone and are evaluated based on UniTrack.

Method	SOT		VOS	MOT		MOTS		PoseTrack
	AUC_{XCorr}	AUC_{DCF}	$\mathcal{J}$-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+GeSSL	51.0 / 54.4	**63.7** / 53.5	**62.3 / 58.3**	**70.3 / 77.3**	**60.5 / 65.0**	68.1 / **69.6**	69.2 / **71.2**	73.7 / 74.4
BYOL+GeSSL	**52.0 / 57.9**	60.5 / 59.0	61.2 / 57.3	68.0 / 77.1	58.2 / 64.4	**73.0** / 68.6	**73.6** / 71.1	**75.0 / 75.8**

Table 10: The cross-domain few-shot learning accuracies ($\pm$95% confidence interval). We transfer models trained on miniImageNet to six benchmark datasets with the C4-backbone. The best results are highlighted in **bold**. The (N, A) means the N-way A-shot tasks with N classes and $N \times A$ samples, where each class has A samples augmented from the same image.

Method	**CUB**		**Cars**		**Places**	
	(5,5)	**(5,20)**	**(5,5)**	**(5,20)**	**(5,5)**	**(5,20)**
Unsupervised Few-shot Learning						
MetaSVEBM	45.893 $\pm$ 0.334	54.823 $\pm$ 0.347	33.530 $\pm$ 0.367	44.622 $\pm$ 0.299	50.516 $\pm$ 0.397	61.561 $\pm$ 0.412
MetaGMVAE	48.783 $\pm$ 0.426	55.651 $\pm$ 0.367	30.205 $\pm$ 0.334	39.946 $\pm$ 0.400	55.361 $\pm$ 0.237	65.520 $\pm$ 0.374
PsCo	56.365 $\pm$ 0.636	69.298 $\pm$ 0.523	44.632 $\pm$ 0.726	56.990 $\pm$ 0.551	64.501 $\pm$ 0.780	73.516 $\pm$ 0.499
Self-supervised Learning						
SimCLR	51.389 $\pm$ 0.365	60.011 $\pm$ 0.485	38.639 $\pm$ 0.432	52.412 $\pm$ 0.783	59.523 $\pm$ 0.461	68.419 $\pm$ 0.500
MoCo	52.843 $\pm$ 0.347	61.204 $\pm$ 0.429	39.504 $\pm$ 0.489	50.108 $\pm$ 0.410	60.291 $\pm$ 0.583	69.033 $\pm$ 0.654
SwAV	51.250 $\pm$ 0.530	61.645 $\pm$ 0.411	36.352 $\pm$ 0.482	51.153 $\pm$ 0.399	58.789 $\pm$ 0.403	68.512 $\pm$ 0.466
SimCLR + GeSSL	55.922 $\pm$ 0.471	64.723 $\pm$ 0.214	43.892 $\pm$ 0.198	56.100 $\pm$ 0.269	65.125 $\pm$ 0.301	72.892 $\pm$ 0.240
MoCo + GeSSL	**57.650 $\pm$ 0.221**	65.502 $\pm$ 0.274	**45.529 $\pm$ 0.295**	55.354 $\pm$ 0.237	**66.602 $\pm$ 0.180**	**74.126 $\pm$ 0.243**
SwAV + GeSSL	55.421 $\pm$ 0.173	**65.927 $\pm$ 0.460**	42.237 $\pm$ 0.296	**56.682 $\pm$ 0.380**	64.601 $\pm$ 0.325	72.460 $\pm$ 0.463

Method	**CropDiseases**		**ISIC**		**ChestX**	
	(5,5)	**(5,20)**	**(5,5)**	**(5,20)**	**(5,5)**	**(5,20)**
Unsupervised Few-shot Learning						
MetaSVEBM	71.652 $\pm$ 0.837	84.515 $\pm$ 0.902	37.106 $\pm$ 0.732	48.001 $\pm$ 0.723	27.238 $\pm$ 0.685	29.652 $\pm$ 0.610
MetaGMVAE	72.683 $\pm$ 0.527	80.777 $\pm$ 0.511	30.630 $\pm$ 0.423	37.574 $\pm$ 0.399	24.522 $\pm$ 0.405	26.239 $\pm$ 0.422
PsCo	**89.565 $\pm$ 0.372**	95.492 $\pm$ 0.399	43.632 $\pm$ 0.400	54.886 $\pm$ 0.359	21.907 $\pm$ 0.258	24.182 $\pm$ 0.389
Self-supervised Learning						
SimCLR	80.360 $\pm$ 0.488	89.161 $\pm$ 0.456	44.669 $\pm$ 0.510	51.823 $\pm$ 0.411	26.556 $\pm$ 0.385	30.982 $\pm$ 0.422
MoCo	81.606 $\pm$ 0.485	90.366 $\pm$ 0.377	44.328 $\pm$ 0.488	52.398 $\pm$ 0.396	24.198 $\pm$ 0.400	27.893 $\pm$ 0.412
SwAV	80.055 $\pm$ 0.502	89.917 $\pm$ 0.539	43.200 $\pm$ 0.356	50.109 $\pm$ 0.350	21.252 $\pm$ 0.439	28.270 $\pm$ 0.417
SimCLR + GeSSL	84.526 $\pm$ 0.413	94.572 $\pm$ 0.332	**47.310 $\pm$ 0.389**	55.710 $\pm$ 0.312	**30.876 $\pm$ 0.259**	**34.492 $\pm$ 0.398**
MoCo + GeSSL	85.852 $\pm$ 0.358	**95.540 $\pm$ 0.335**	46.437 $\pm$ 0.339	**56.466 $\pm$ 0.270**	29.216 $\pm$ 0.332	31.545 $\pm$ 0.279
SwAV + GeSSL	85.355 $\pm$ 0.327	94.785 $\pm$ 0.339	46.521 $\pm$ 0.288	55.268 $\pm$ 0.312	27.462 $\pm$ 0.340	32.237 $\pm$ 0.199

model (Guo et al., 2020). To ensure that GeSSL can achieve robust performance in real-world applications, we further conduct comparative experiments on cross-domain few-shot learning.

Experimental setup. We compare our proposed GeSSL with the few-shot learning baselines as described in Table 4 on cross-domain few-shot learning. We adopt six cross-domain few-shot learning benchmark datasets, and divide these datasets into two categories according to their similarity with ImageNet: i) high similarity: CUB (Welinder et al., 2010), Cars (Krause et al., 2013), and Places (Zhou et al., 2017); ii) low similarity: CropDiseases (Mohanty et al., 2016), ISIC (Codella et al., 2018), and ChestX (Wang et al., 2017). The (N, A) in the tables means the N-way A-shot tasks with N classes and $N \times A$ samples, where each class has A samples augmented from the same image.

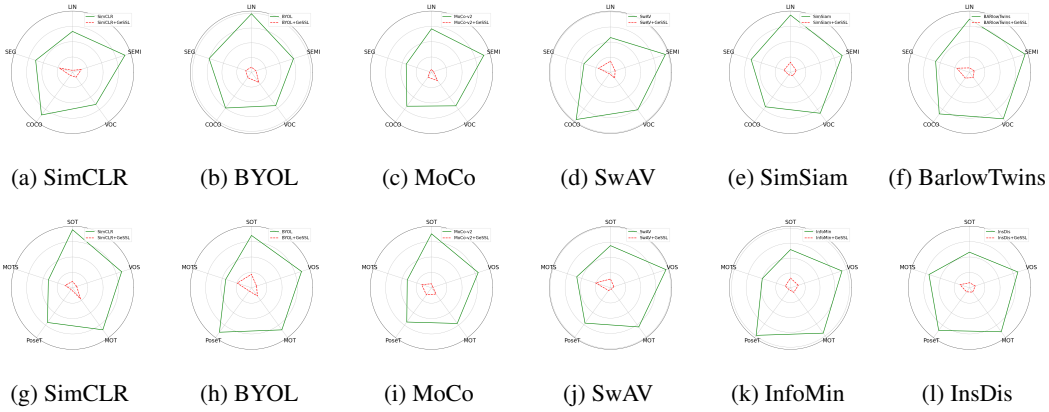
Results. Table 10 presents the performance of the model trained on miniImageNet and transferred to the six cross-domain few-shot learning benchmark datasets mentioned above. By observation, we further validate the performance of our proposed GeSSL: i) Effectiveness: achieves better results than the state-of-the-art baselines on almost all benchmark datasets; ii) Generalization: achieves nearly a

Table 11: Top-1 validation accuracy on ImageNet-1K dataset for ViT-B and ViT-L.

Method	Epoch	ViT-B	ViT-L
data2vec 2.0	200/150	80.5	81.8
data2vec 2.0 + GeSSL	200/150	85.9	88.2

Table 12: Downstream classification accuracy of SimCLR-SAS on CIFAR-10.

Method	Subset Size	Top-1 Accuracy (%)
SimCLR-SAS	10%	79.7
SimCLR-SAS + GeSSL	10%	84.1



(a) SimCLR (b) BYOL (c) MoCo (d) SwAV (e) SimSiam (f) BarlowTwins

(g) SimCLR (h) BYOL (i) MoCo (j) SwAV (k) InfoMin (l) InsDis

Figure 5: Performance of different models on five image-based tasks (top row) and five video-based tasks (bottom row). We choose $\sigma-$measure as the measurement. It is worth noting that the smaller the $\sigma-$measurefen score, the better the effect. Meanwhile, we normalize the results of $\sigma-$measurefen scores on different datasets and compare the performance between baselines by comparing the corresponding branch of the fan chart.

3% improvement compared to unsupervised few-shot Learning and self-supervised learning on the datasets with significant differences from the training phase; iii) Robustness: achieves better results than the PsCo (Jang et al., 2023) which introduces out-of-distribution samples, even though we do not explicitly consider out-of-distribution samples on datasets with significant differences.

C.4 Evaluation of Distribution-level Consistency

Current SSL models often construct relations based on spurious correlations, making it challenging for them to remain stable across varied downstream environments (Huang et al., 2021; Sun et al., 2020; Ericsson et al., 2022). As discussed in our methodology, GeSSL explicitly approximates counterfactual sufficiency to encourage distribution-level consistency. In this section, we quantitatively evaluate whether GeSSL helps the learned representations remain stable and reliable across diverse tasks compared to existing SSL methods.

Specifically, we introduce a stability measure (σ-measure) to assess the difference in performance between the learned model and an optimal semantic oracle for various empirical environments. The optimal model is assumed to output the ground-truth semantic class, and the deviation is quantified using the KL divergence between the predicted and true class probability distributions. A lower score indicates that the self-supervised relations align better with true semantic factors (Z_c) and are robust to nuisance variations (Z_s).

Specifically, the σ-measurement score assesses the difference in performance between the learned model and the optimal model for each task. The optimal model is assumed to output the ground truth, and the performance difference is quantified using the KL divergence between the predicted and true class probability distributions. It compares the predicted class probabilities produced by classifier π to the true labels across SSL tasks, such as comparing the predicted values $[0.81, 0.09, 0.03, 0.07]$ to the true labels $[1, 0, 0, 0]$. Take the LIN task with SimCLR as an example: we train SimCLR and SimCLR+GeSSL on the COCO dataset for 200 epochs, then add an MLP after the feature extractor. A new mini-batch is input into both SimCLR and SimCLR+GeSSL to generate class probability distributions for each sample, and the KL divergence between these predicted and true distributions

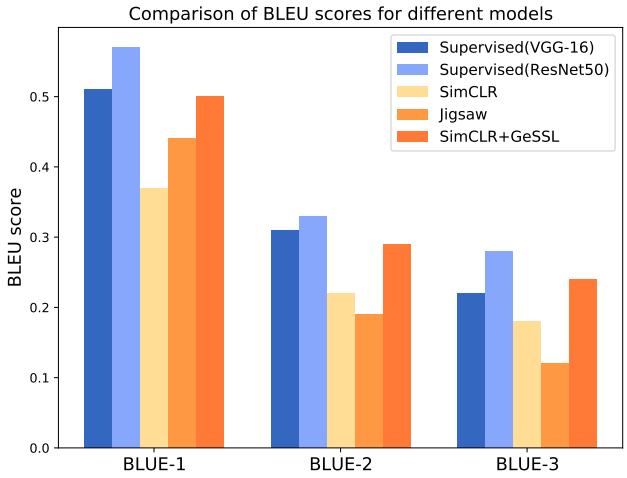


Figure 6: Comparison of BLEU scores for different models, comparing 2 fully supervised and 3 self-supervised pre-text tasks, trained on the Flickr8k.

Table 13: The performance of introducing GeSSL during training. All results are recorded during training using the σ-measurement.

Metric	Training Epochs									
	20	40	60	80	100	120	140	160	180	200
Accuracy of SimCLR	20.1	43.6	51.2	60.2	70.3	77.2	82.3	86.1	88.7	88.6
Accuracy of SimCLR + GeSSL	42.4	67.1	83.0	92.9	93.0	94.4	94.1	93.2	94.1	94.2
Performance Ratio r	0.474	0.650	0.617	0.648	0.756	0.818	0.875	0.924	0.943	0.941

is calculated. After normalization, the scores for the LIN task are obtained, with similar evaluations conducted for other baselines and tasks.

In the experiments, we chose two scenarios based on images and videos to evaluate the model's versatility following (Liu et al., 2022b). The image-based tasks include linear probing (top-1 accuracy) with 800-epoch pre-trained models (LIN), semi-supervised classification (top-1 accuracy) using a 1% subset of training data (SEMI), object detection (AP) on VOC dataset (VOC) and COCO dataset (COCO), and instance segmentation (AP^{mask}) on COCO dataset (SEG). For video-based tasks, we compute rankings in terms of AUC for SOT, $\mathcal{J}$-mean for VOS, IDF-1 for MOT, IDF-1 for PoseTracking, and IDF-1 for MOTS, respectively. Next, we evaluate the σ-measurement scores of different baselines before and after the introduction of GeSSL and after training for 200 epochs. Among them, the better model is set to the result of ground truth, and the calculation of σ-measurement score is performed on a series of randomly sampled tasks. Figure 5 shows the comparison results. The lower $\sigma-$measure denotes the better performance. We can observe that: (i) the σ-measurement score of the existing SSL model is low and it is difficult to achieve good results in multiple domains and tasks; (ii) after the introduction of GeSSL, the σ-measurement score of the SSL models are significantly decreased.

Besides, in the first 20-200 epochs of training (each epoch contains multiple tasks), we evaluate the average performance of multiple f' in each epoch. Each f'_θ is obtained by updating f_θ on the corresponding support set. We calculate the accuracy of SimCLR before and after the introduction of GeSSL and the ratio r of their effects on the CIFAR-10 data set. If $r < 1$, it means that the representation effect learned by the model in each epoch of training is better when introducing GeSSL. The results for every 20 epochs are shown in Table 13. The results show that: (i) r is always less than 1, which proves that the representation effect learned after the introduction of GeSSL is significantly improved; (ii) after the introduction of GeSSL, the accuracy of the model is significantly improved, and it becomes stable after 80 epochs, i.e., great results can be achieved for even based on just one iteration and few data. These results show that "the model f_θ achieves comparable performance on each task quickly with few data during training" after introducing GeSSL.

Table 14: Comparison between models.

Method	scratch, original	scratch, our impl.	baseline MAE	MAE + Our
Top 1	76.5	82.7	85.3	88.1

Table 15: Comparisons with previous results on ImageNet-1K. The ViT models are B/16, L/16, H/14 (Dosovitskiy et al., 2020). The pre-training data is the ImageNet-1K training set (except the tokenizer in BEiT was pre-trained on 250M DALLE data (Ramesh et al., 2021)). All results are on an image size of 224, except for ViT-H with an extra result of 448.

Method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H_{448}
DINO	IN1K	82.8	-	-	-
MoCo	IN1K	83.2	84.1	-	-
BEiT	IN1K+DALLE	83.2	85.2	-	-
MAE	IN1K	83.6	85.9	86.9	87.8
MAE+Ours	IN1K	87.6	88.5	89.2	89.7

C.5 Evaluation on Generative Self-supervised Learning

In this Section, we evaluate the effectiveness of the proposed GeSSL on the generative self-supervised learning paradigm. We conduct experiments on three scenarios, including image generation, image captioning, and object detection and segmentation.

Evaluation on Image Generation. To explore the effect of GeSSL on generative SSL, we conduct a set of experiments on ImageNet-1K dataset (Deng et al., 2009b). Specifically, we begin by conducting self-supervised pre-training on the ImageNet-1K (IN1K) training set. Following this, we carry out supervised training to assess the representations using either (i) end-to-end fine-tuning or (ii) linear probing. The results are reported as the top-1 validation accuracy for a single 224×224 crop. For this process, we utilize ViT-Large (ViT-L/16) (Dosovitskiy et al., 2020) as the backbone. Note that ViT-L is very big (an order of magnitude bigger than ResNet-50 (He et al., 2016)) and tends to overfit, as shown in Table 14. The comparison results are shown in Table 15. We can observe that GeSSL achieves stable performance improvements

Evaluation on Image Captioning. We use the commonly used protocol following (Mohamed et al., 2022). The dataset we use to train the pretext task is the unlabeled part of MSCOCO dataset (Vinyals et al., 2016b), which contains 123K images with an average resolution of 640×480 pixels. This dataset contains color and grayscale images. For downstream tasks, we use the Flicker8K dataset (Hodosh et al., 2013). Next, we train it using pre-trained pre-text tasks supervised by VGG-16 and ResNet-50, as well as self-supervised pre-text tasks from SimCLR and Jigsaw Puzzle solutions. In the next step, to evaluate the results, we use the BLEU (Bilingual Evaluation Research) score as the evaluation metric, which evaluates the generated sentences against the reference sentences, where a perfect match is 1 and a perfect mismatch is 0, calculating scores for 1, 2, 3 and 4 cumulative n-grams. The results are shown in Figure 6. From the results, we can observe that after introducing the GeSSL framework we proposed, the model effect has been further improved, stably exceeding the SOTA of the SSL method, and even approaching the supervised learning results. The results show that our proposed GeSSL can still achieve good results in generative self-supervised learning.

Evaluation on Object Detection and Segmentation. For object detection and segmentation, we fine-tune Mask R-CNN (He et al., 2017) end-to-end on COCO (Lin et al., 2014b). The ViT backbone is adapted for use with FPN (Lin et al., 2017). We report box AP for object detection and mask AP for instance segmentation. The results are shown in Table 16. Compared to supervised pre-training, our MAE performs better under all configurations. Our method still achieves optimal results, demonstrating its effectiveness.

C.6 Evaluation on More Modalities

GeSSL proposed in this work can be applied in various fields and domains, e.g., instance segmentation, video tracking, sample generation, etc., as mentioned before. Here, we provide the experiments of

Table 16: COCO object detection and segmentation using a ViT Mask R-CNN baseline. All self-supervised entries use IN1K data without labels, and Mask AP follows a similar trend as box AP.

Method	pre-train data	AP^{box}		AP^{mask}	
		ViT-B	ViT-L	ViT-B	ViT-L
supervised	IN1K w/ labels	47.9	49.3	42.9	43.9
MoCo v3	IN1K	47.9	49.3	42.7	44.0
BEiT	IN1K+DALLE	49.8	53.3	44.4	47.1
MAE	IN1K	50.3	53.3	44.9	47.2
MAE + Our	IN1K	54.9	57.3	47.9	53.0

Table 17: Performance on for text recognition.

Methods	**IIIT5K**	**IC03**
SimCLR (Chen et al., 2020a)	1.7	3.8
SeqCLR (Aberdam et al., 2021)	35.7	43.6
SimCLR + GeSSL	21.4	20.8
SeqCLR + GeSSL	41.3	50.6

GeSSL on text modality-based datasets, i.e., IC03 and IIIT5K (Yasmeen et al., 2020), which we have conducted before. We follow the same experimental settings as mentioned in (Aberdam et al., 2021). The results shown in Table 17 demonstrate that GeSSL achieves stable effectiveness and robustness in various modalities combined with the above experiments.

C.7 Visualization Analyses

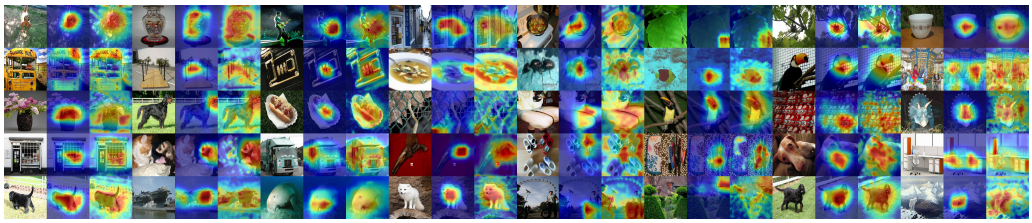
To understand how GeSSL improves the quality of learned representations, we conduct a visualization analyses. Specifically, we compare the baseline self-supervised models, SimCLR and MAE, with their GeSSL-enhanced counterparts. For visualization, we employ Grad-CAM (Selvaraju et al., 2017) and evaluate on randomly selected validation images from miniimageNet. As illustrated in **Figure 7**, the original SimCLR and MAE models may struggle to decouple the primary object from the background. In contrast, when equipped with GeSSL, both SimCLR and MAE demonstrate significantly more concentrated and precise attention on the foreground objects. This phenomenon provides empirical evidence for our theoretical motivation. In the framework of causal relation certification, backgrounds and styles act as nuisance factors (Z_s), which vary across instances and environments. By approximating counterfactual necessity and sufficiency, GeSSL explicitly penalizes reliance on these unstable spurious relations.

D Details of Ablation Study

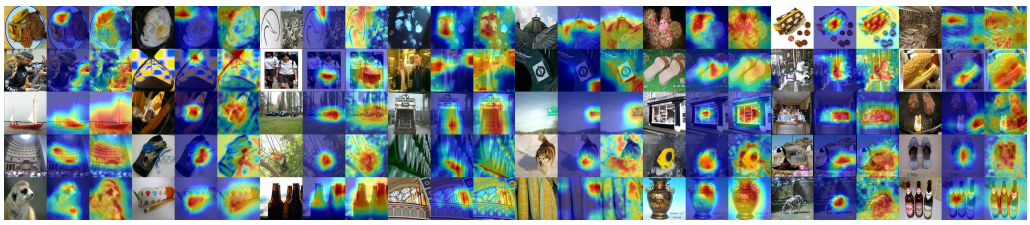
In this section, we introduce the experimental details and more comprehensive analysis of the ablation studies (Subsection 5.2).

D.1 Model Efficiency

This ablation study explores the efficiency of self-supervised models before and after applying GeSSL. Specifically, we choose five baselines, including SimCLR (Chen et al., 2020a), MOCO (Chen et al., 2020b), BYOL (Grill et al., 2020), Barlow Twins (Zbontar et al., 2021), and SwAV (Caron et al., 2020b). Then, we evaluate the accuracy, training hours, and parameter size of these models on STL-10 before and after applying our proposed GeSSL. We use the same linear evaluation setting as in Section 5.1 of the main text. Finally, we plot the trade-off scatter plot by recording the average values of five runs. The results are shown in Figure 2 of the main text, where the horizontal axis represents the training hours and the vertical axis represents the accuracy. The center of each circle represents the result of the training time and accuracy of each model, and the area of the circle represents the parameter size. The numerical results of this experiment are shown in Table 19. From the results, we can see that GeSSL can significantly improve the performance and computational efficiency of self-supervised learning models.



(a) SimCLR vs. SimCLR+Ours



(b) MAE vs. MAE+Ours

Figure 7: Visualization Analyses.

Table 18: Training cost per epoch of SSL models.

Methods	Training Cost per Epoch (s)
SimCLR (Chen et al., 2020a)	12.8
MOCO (Chen et al., 2020b)	16.9
SimCLR + GeSSL	9.6
MOCO + GeSSL	12.0

Note that although the optimization method used by GeSSL is more complex, one of its core goals is to accelerate model convergence, i.e., achieve greater performance improvement per unit of time. This does not imply that GeSSL always requires fewer epochs to reach the optimal result. In fact, GeSSL uses approximate implicit differentiation with finite difference (AID-FD) for updates instead of conventional explicit second-order differentiation (as mentioned in Appendix D.4). Moreover, GeSSL constructs a self-motivated target that guides the model to optimize more effectively in a specific task. Therefore, the efficiency improvement is reflected in the computational efficiency and effectiveness of updates per epoch, rather than simply reducing the total number of epochs. Furthermore, to verify whether the efficiency improvement is attributable to a single epoch, we separately measured the computational overhead of SSL baseline algorithms after integrating GeSSL for a single epoch. The results, presented in Table 18, demonstrate that with a consistent batch size, GeSSL enhances the computational efficiency and the effectiveness of updates per epoch for the SSL baseline algorithms.

D.2 Ablation Study of $\mathcal{L}_{disc}$

To evaluate the impact of $\mathcal{L}_{\text{disc}}$, we conduct an ablation study analyzing its effect on the learned representation space. In GeSSL, $\mathcal{L}_{\text{disc}}$ approximates counterfactual necessity by learning an adaptive relation boundary that suppresses spurious relations and enhances discriminability. We visualize representations before and after incorporating $\mathcal{L}_{\text{disc}}$ across four representative baselines, SimCLR, BYOL, MoCo, and MAE, using 10 randomly sampled classes from ImageNet-100 (100 samples per class). As shown in Figure 8, introducing $\mathcal{L}_{\text{disc}}$ yields markedly sharper class boundaries and prevents distinct semantic categories from collapsing into overlapping embeddings, confirming its effectiveness in enforcing counterfactual necessity.

D.3 More Experiments of Parameter Sensitivity

Considering that our framework updates the self-supervised model f_θ in GeSSL based on M relation environments simultaneously, the number of sampled samples per batch of self-supervised learning directly determines the class diversity of the data in the mini-batch. In this section, we further conduct ablation experiments on the number of pairs within each batch and the batch size that are learned simultaneously.

Table 19: Model analysis including parameter size, training time, and performance.

Methods	Memory Footprint (MiB)	Parameter Size (M)	Training Time (h)	Accuracy (%)
SimCLR	2415	23.15	4.15	90.5
MOCO	2519	24.01	4.96	90.9
BYOL	2691	25.84	6.98	91.9
BarlowTwins	2477	23.15	5.88	90.3
SwAV	2309	22.07	4.45	90.7
SimCLR+GeSSL	2784	26.21	3.36	93.4
MOCO+GeSSL	2912	27.20	4.23	94.6
BYOL+GeSSL	2875	28.01	5.70	94.8
BarlowTwins+GeSSL	2856	27.11	5.39	94.2
SwAV+GeSSL	3012	28.61	3.96	93.2

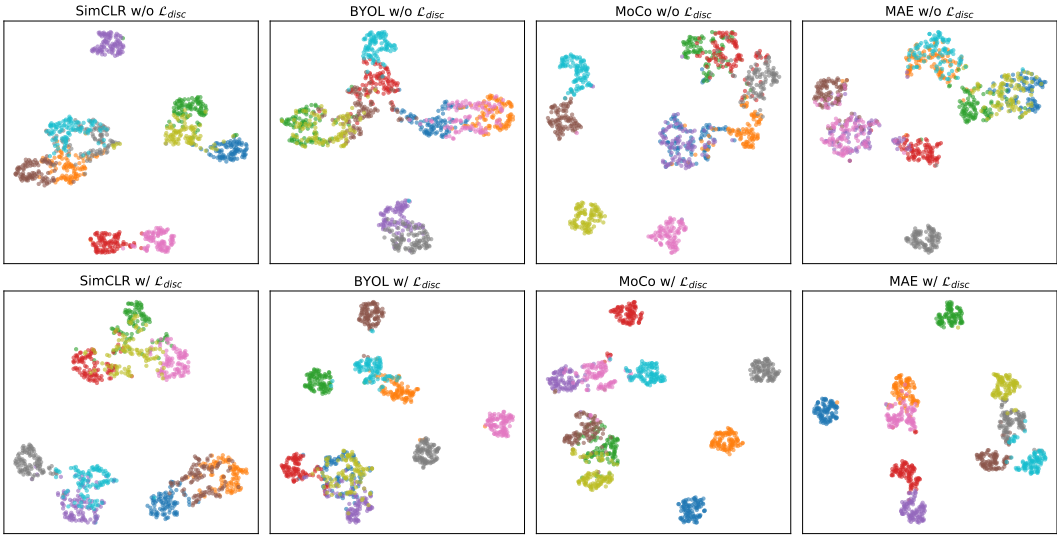


Figure 8: Ablation study of $\mathcal{L}_{disc}$. We perform t-SNE visualization to evaluate the classification performance of the SSL model before and after introducing $\mathcal{L}_{disc}$.

Specifically, we choose the commonly used STL-10 for unsupervised learning, ImageNet with 10% label for semi-supervised learning, and miniImageNet for few-shot learning, and evaluate the performance of SimCLR + GeSSL under different batch sizes and different n values. Figure 9 shows the impact of different number of pairs for SSL. The results show that SimCLR + GeSSL always outperforms SimCLR under any batch size. A larger batch size leads to a slightly larger performance improvement for SimCLR + GeSSL, but also increases the computational resource consumption. Therefore, in this study, we build relation environments based on images with a batch size of $n = 16$ or $n = 32$. Figure 10 shows the impact of the batchsize for the outer-loop optimization. The results indicate that $m = 8$ is a better trade-off between model accuracy and time consumption. In the setting of our GeSSL, we also choose $m = 8$ as the hyperparameter setting.

In addition, considering that GeSSL updates every m mini-batches, we evaluate the baseline performance under $m\times$ the original batch size. Specifically, we adopt the same experimental setup as in Figure 2, with the only difference being that we increase the batch size of the SimCLR baseline by a factor of m and record the results. The results are shown in Table 20, which indicates that the

Table 20: Performance on for a large batchsize.

Methods	Accuracy	Training Cost
SimCLR (Chen et al., 2020a)	90.8	5.2
SimCLR + GeSSL	93.8	3.9

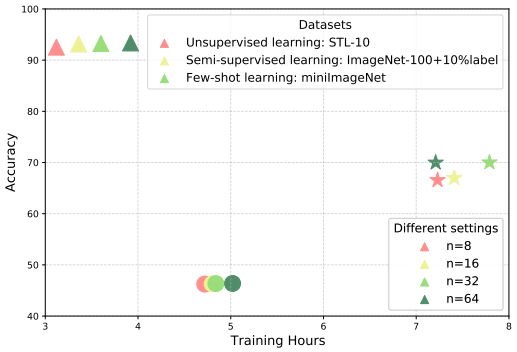


Figure 9: Ablation study of the number of pairs.

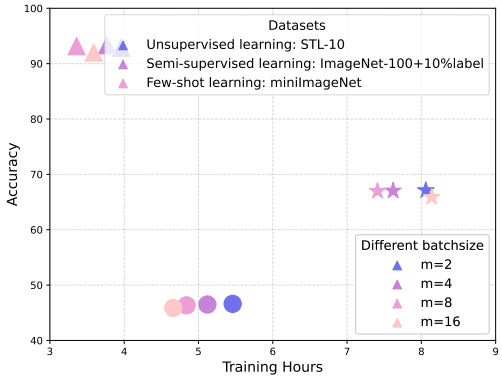


Figure 10: Ablation study of the batchsize.

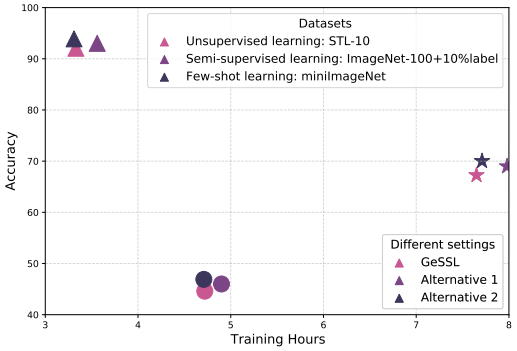


Figure 11: Evaluation of bi-level optimization.

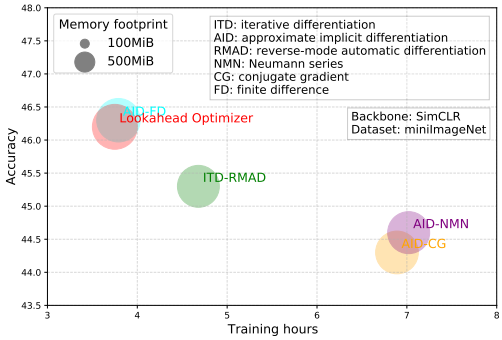


Figure 12: Implementation of optimization.

performance of SimCLR, after converging with the larger training data, remains largely unchanged and still inferior to GeSSL.

D.4 Evaluation and Implementation of the Bi-level Optimization

As mentioned in Subsection 5.1, to assess the advantages of our bi-level optimization, we compare its performance against two alternatives: (i) jointly optimizing the inner and outer objectives in a single stage; and (ii) training a distinct f' for each mini-batch. The results in Figure 11 demonstrate that our bi-level optimization (Subsection 3.3) achieves state-of-the-art performance.

The model of GeSSL is updated based on bi-level optimization, and the model gradients for each level are obtained by combining the optimal response Jacobian matrices through the chain rule. In practical applications, multi-level gradient computation requires a lot of memory and computation (Choe et al., 2022), so we hope to introduce a more concise gradient backpropagation and update method to reduce the computational complexity. Specifically, we consider two types of gradient update methods, including iterative differentiation (ITD) (Finn et al., 2017a) and approximate implicit differentiation (AID) (Grazzi et al., 2020). We provide implementations of four popular ITD/AID algorithms, including ITD with reverse-mode automatic differentiation (ITD-RMAD) (Finn et al., 2017a), AID with Neumann series (AID-NMN) (Lorraine et al., 2020), AID with conjugate gradient (AID-CG) (Rajeswaran et al., 2019), and AID with finite difference (AID-FD) (Liu et al., 2018). We also choose the recently proposed optimizer, i.e., Lookahead (Zhang et al., 2019) for comparison. We denote the the upper-level parameters and the lower-level parameters as θ and ϕ, respectively. All the way of gradient update of the bi-level optimization are as follows:

ITD-RMAD (Finn et al., 2017a), ITD with reverse-mode automatic differentiation applies the implicit function theorem to the lower-level optimization problem and computes the gradients of the upper-level objective with respect to the upper-level parameters using reverse-mode automatic differentiation. The update process is as follows:

- Solve the lower-level optimization problem $\phi^* = \arg\min_\phi L(\phi, \theta)$ using gradient descent.
- Compute the gradient of the upper-level objective $g(\theta) = F(\phi^*, \theta)$ with respect to θ using reverse-mode automatic differentiation:

$$\nabla_\theta g(\theta) = \nabla_\theta F(\phi^*, \theta) - \nabla_\phi F(\phi^*, \theta)^T (\nabla_\phi L(\phi^*, \theta))^{-1} \nabla_\theta L(\phi^*, \theta) \tag{16}$$

- Update the upper-level parameters using gradient descent or other methods: $\theta \leftarrow \theta - \alpha \nabla_\theta g(\theta)$.

AID-NMN (Lorraine et al., 2020), AID with Neumann series, approximates the inverse of the Hessian matrix of the lower-level objective using a truncated Neumann series expansion and computes the gradients of the upper-level objective with respect to the upper-level parameters using forward-mode automatic differentiation. The update process is as follows:

- Solve the lower-level optimization problem $\phi^* = \arg\min_\phi L(\phi, \theta)$ using gradient descent.
- Compute the gradient of the upper-level objective $g(\theta) = F(\phi^*, \theta)$ with respect to θ using forward-mode automatic differentiation:

$$\begin{aligned} &\nabla_\theta g(\theta) = \nabla_\theta F(\phi^*, \theta) - \nabla_\phi F(\phi^*, \theta)^T (\nabla_\phi L(\phi^*, \theta))^{-1} \nabla_\theta L(\phi^*, \theta) \\ &\approx \nabla_\theta F(\phi^*, \theta) - \nabla_\phi F(\phi^*, \theta)^T \textstyle\sum_{k=0}^{K} (-1)^k (\nabla_\phi^2 L(\phi^*, \theta))^k \nabla_\theta L(\phi^*, \theta) \end{aligned} \tag{17}$$

where K is the truncation order of the Neumann series.

- Update the upper-level parameters using gradient descent or other methods: $\theta \leftarrow \theta - \alpha \nabla_\theta g(\theta)$.

AID-CG (Rajeswaran et al., 2019), AID with conjugate gradient, solves a linear system involving the Hessian matrix of the lower-level objective using the conjugate gradient algorithm and computes the gradients of the upper-level objective with respect to the upper-level parameters using forward-mode automatic differentiation. The update process is as follows:

- Solve the lower-level optimization problem $\phi^* = \arg\min_\phi L(\phi, \theta)$ using gradient descent or other methods.
- Compute the gradient of the upper-level objective $g(\theta) = F(\phi^*, \theta)$ with respect to θ using forward-mode automatic differentiation:

$$\begin{aligned} &\nabla_\theta g(\theta) = \nabla_\theta F(\phi^*, \theta) \\ &-\nabla_\phi F(\phi^*, \theta)^T (\nabla_\phi L(\phi^*, \theta))^{-1} \nabla_\theta L(\phi^*, \theta) \approx \nabla_\theta F(\phi^*, \theta) \\ &-\nabla_\phi F(\phi^*, \theta)^T v \end{aligned} \tag{18}$$

where v is the solution of the linear system $(\nabla_\phi^2 L(\phi^*, \theta))v = \nabla_\theta L(\phi^*, \theta)$ obtained by the conjugate gradient algorithm.

- Update the upper-level parameters using gradient descent or other methods: $\theta \leftarrow \theta - \alpha \nabla_\theta g(\theta)$.

AID-FD (Liu et al., 2018), AID with finite difference, approximates the inverse of the Hessian matrix of the lower-level objective using a finite difference approximation and computes the gradients of the upper-level objective with respect to the upper-level parameters using forward-mode automatic differentiation. The update process is as follows:

- Solve the lower-level optimization problem $\phi^* = \arg\min_\phi L(\phi, \theta)$ using gradient descent or other methods.
- Compute the gradient of the upper-level objective $g(\theta) = F(\phi^*, \theta)$ with respect to θ using forward-mode automatic differentiation:

$$\begin{aligned}
&\nabla_\theta g(\theta) = \nabla_\theta F(\phi^*, \theta) \\
&-\nabla_\phi F(\phi^*, \theta)^T (\nabla_\phi L(\phi^*, \theta))^{-1} \nabla_\theta L(\phi^*, \theta) \\
&\approx \nabla_\theta F(\phi^*, \theta) \\
&-\nabla_\phi F(\phi^*, \theta)^T \frac{\nabla_\theta L(\phi^* + \epsilon \nabla_\theta L(\phi^*, \theta), \theta) - \nabla_\theta L(\phi^*, \theta)}{\epsilon}
\end{aligned} \tag{19}$$

where ϵ is a small positive constant for the finite difference approximation.

- Update the upper-level parameters using gradient descent or other methods: $\theta \leftarrow \theta - \alpha \nabla_\theta g(\theta)$.

Lookahead (Zhang et al., 2019) introduces a novel approach to optimization by maintaining two sets of weights: the fast and the slow weights. The fast weights, θ_{fast}, are updated frequently through standard optimization techniques, while the slow weights, θ_{slow}, are updated at a lesser frequency. The key formula that updates the slow weights is given by:

$$\theta_{\text{slow}} \leftarrow \theta_{\text{slow}} + \alpha(\theta_{\text{fast}} - \theta_{\text{slow}}) \tag{20}$$

where α is a hyperparameter controlling the step size. This method aims to stabilize training and ensure consistent convergence.

The results shown in Figure 4 of the main text demonstrate that approximate implicit differentiation with finite difference also achieves optimal results on the SSL model. Our optimization process is also based on this setting.

D.5 Robustness to Augmentation Reliance.

Standard D-SSL methods heavily rely on hand-crafted augmentation compositions (e.g., color jitter). To test whether GeSSL genuinely certifies underlying semantic relations rather than relying on specific augmentations, we conduct progressive augmentation-stripping experiments. The results are shown in Table 21. GeSSL exhibits remarkable resilience even under the extreme crop-only regime, outperforming baselines by over 12% on CIFAR-10 and 15% on ImageNet-100.

Table 21: Top-1 linear probing accuracy (%) on CIFAR-10 and ImageNet-100 under progressive augmentation stripping (ResNet-18 / ResNet-50, 200 epochs).

(a) Ablation on CIFAR-10 across 5 stripping levels

Augmentation	SimCLR	+ GeSSL	MoCo-v2	+ GeSSL
Full Pipeline	90.5	93.4	90.6	94.1
w/o Blur	89.2	92.8	89.5	93.3
w/o Blur & Gray	86.1	90.7	86.8	91.2
w/o Blur, Gray, Jitter	74.3	82.1	75.6	83.5
Crop Only	62.4	74.8	63.8	76.1

(b) Ablation on ImageNet-100

Backbone	Method	Augmentation	Top-1 (%)
ResNet-50	SimCLR	Full Pipeline	70.15
	SimCLR	Crop Only	48.20
	SimCLR + GeSSL	Full Pipeline	**72.96**
	SimCLR + GeSSL	Crop Only	**63.50**
ViT-B/16	MAE	Full Pipeline	76.56
	MAE + GeSSL	Crop Only	**71.20**

D.6 Detailed Computation Complexity and Convergence Trade-off

Although GeSSL incorporates bi-level optimization, it avoids explicitly computing or inverting the second-order Hessian matrix by adopting AID-FD. Computing the outer gradient with respect to ϕ only introduces one auxiliary finite-difference forward/backward evaluation:

$$\nabla_\phi \mathcal{L}_{que} \approx -\nabla_\phi F(\phi^*, \theta)^T \frac{\nabla_\theta L(\phi^* + \epsilon \nabla_\theta L(\phi^*, \theta), \theta) - \nabla_\theta L(\phi^*, \theta)}{\epsilon}. \tag{21}$$

To quantify the exact computational overhead, we profile the per-step theoretical FLOPs, GMACs, and peak memory footprint alongside overall training efficiency in Table 22. From the results, we can observe that AID-FD incurs only a marginal per-step compute increase (≈19.5% FLOPs and 15% memory). GeSSL's relation certification acts as an informative regularizer that substantially accelerates convergence: reaching target performance in roughly 80 epochs, compared to 200 epochs required by the vanilla counterpart. This results in an overall reduction in total wall-clock training time.

Table 22: Per-step computation complexity, memory footprint, and wall-clock convergence comparison.

Method	GFLOPs	GMACs	Memory (MiB)	Training Time
SimCLR	8.24	4.11	2,415	4.15 h (200 ep)
SimCLR + GeSSL	9.88 (+19.9%)	4.92 (+19.7%)	2,784 (+15.2%)	3.36 h (80 ep)
MoCo-v2	8.52	4.25	2,519	4.96 h (200 ep)
MoCo-v2 + GeSSL	10.15 (+19.1%)	5.06 (+19.1%)	2,912 (+15.6%)	4.23 h (80 ep)

E Additional Analysis of Counterfactual Relation Certification

This section provides additional analyses of the proposed counterfactual relation certification framework. The goal is not to directly observe true counterfactual relations, since true interventions and latent semantic factors are unavailable in SSL. Instead, we examine whether the proposed observable proxies behave consistently with our formulation. Specifically, we analyze relation reliability, robustness under nuisance shifts, the contribution of each certification proxy, and the correlation between the certification objective and downstream performance.

E.1 Relation Reliability Analysis

The discriminative relation loss is designed as a proxy for counterfactual necessity. It should help reject relations that are supported only by superficial similarity and preserve relations that are more likely to be semantically compatible. To evaluate this behavior, we measure the reliability of accepted and rejected relations after pre-training.

For each anchor x_i^{anc}, GeSSL predicts an anchor-specific relation boundary a_i. For a candidate sample x_j, we compute the relation score

$$s_{ij} = a_i - d(f(x_j), f(x_i^{\mathrm{anc}})),$$

where a larger s_{ij} indicates a more compatible relation. For baseline SSL methods without an adaptive boundary, we use the negative cosine distance,

$$s_{ij} = -d(f(x_j), f(x_i^{\mathrm{anc}})),$$

as the relation score. For fair comparison, all methods accept the same ratio of candidate relations per anchor. Labels are used only for evaluation and are never used during SSL pre-training.

We report three metrics. Accepted Purity measures the fraction of accepted relations whose samples share the same class label as the anchor:

$$\mathrm{Purity} = \frac{\#\{(i,j) : \mathrm{accept}(i,j), y_i = y_j\}}{\#\{(i,j) : \mathrm{accept}(i,j)\}}.$$

Rejection Accuracy measures the fraction of rejected relations whose samples have different labels:

$$\mathrm{RejectAcc} = \frac{\#\{(i,j) : \mathrm{reject}(i,j), y_i \neq y_j\}}{\#\{(i,j) : \mathrm{reject}(i,j)\}}.$$

Relation AUROC evaluates how well the relation score predicts whether two samples share the same class label.

As shown in Table 23, GeSSL consistently improves relation reliability across both discriminative and generative SSL baselines. This suggests that the adaptive relation boundary improves the quality of the self-supervised relations used during training, rather than only improving downstream accuracy.

E.2 Robustness under Nuisance Shifts

Counterfactual sufficiency requires that a valid relation should remain stable when nuisance factors change while semantic factors are preserved. To evaluate whether GeSSL improves this property, we test frozen representations under common nuisance shifts, including corruption, blur, noise, texture changes, and color perturbations.

Table 23: Relation reliability analysis on ImageNet-100. Labels are used only for evaluation. GeSSL improves the purity of accepted relations and the accuracy of rejected relations, supporting the role of $\mathcal{L}_{disc}$ as a proxy for counterfactual necessity.

Method	Accepted Purity ↑	Rejection Accuracy ↑	Relation AUROC ↑
SimCLR	61.8 ± 0.3	69.2 ± 0.4	72.4 ± 0.3
SimCLR + GeSSL	67.9 ± 0.4	74.8 ± 0.3	78.6 ± 0.4
MoCo	63.4 ± 0.2	70.8 ± 0.3	74.1 ± 0.3
MoCo + GeSSL	69.7 ± 0.3	76.2 ± 0.4	80.1 ± 0.3
MAE	62.1 ± 0.4	69.5 ± 0.4	72.8 ± 0.4
MAE + GeSSL	68.5 ± 0.3	75.6 ± 0.3	79.3 ± 0.3

Table 24: Robustness under nuisance shifts. GeSSL improves corrupted accuracy and reduces the clean-to-corrupted performance drop, indicating stronger relation stability under nuisance variations.

Method	CIFAR-10-C		CIFAR-100-C		ImageNet-100-C	
	Corr. Acc. ↑	Drop ↓	Corr. Acc. ↑	Drop ↓	Corr. Acc. ↑	Drop ↓
SimCLR	75.6 ± 0.4	16.2	43.2 ± 0.5	25.2	49.8 ± 0.4	20.3
SimCLR + GeSSL	80.1 ± 0.3	13.4	48.0 ± 0.4	23.0	54.7 ± 0.4	18.2
MoCo	77.3 ± 0.3	15.4	45.1 ± 0.4	24.6	51.2 ± 0.3	19.6
MoCo + GeSSL	81.5 ± 0.4	12.9	49.6 ± 0.4	22.1	55.8 ± 0.3	17.5
MAE	78.0 ± 0.4	14.8	45.8 ± 0.5	23.8	52.0 ± 0.4	18.9
MAE + GeSSL	82.0 ± 0.3	12.5	50.4 ± 0.4	21.6	56.1 ± 0.3	17.1

We follow the standard linear evaluation protocol. A linear classifier is trained on clean training data with the pre-trained feature extractor frozen. The classifier is then evaluated on corrupted test data. For corrupted datasets, we average accuracy over all corruption types and severity levels. We report corrupted accuracy and the clean-to-corrupted performance drop:

$$\Delta = \mathrm{Acc}_{clean} - \mathrm{Acc}_{corr}. \tag{22}$$

A smaller drop indicates stronger robustness under nuisance changes.

Table 24 shows GeSSL improves robustness under nuisance shifts. The results support the sufficiency side of our formulation: the learned relations are more stable when non-semantic factors vary.

E.3 Ablation of Observable Certification Proxies

GeSSL approximates counterfactual relation certification through several observable proxies. The base SSL objective already provides sample-level consistency. Beyond this basic proxy, GeSSL introduces: (i) a discriminative relation loss for counterfactual necessity; (ii) support-query bi-level optimization for set-level counterfactual sufficiency; and (iii) multi-batch relation environments for distribution-level counterfactual sufficiency.

To evaluate their individual contributions, we conduct ablation studies on ImageNet-100. The baseline and full GeSSL results in Table 25 are kept identical to the corresponding ImageNet-100 results in the main paper. This ensures consistency between the main text and the appendix. The intermediate variants are additional ablations and are not duplicated in the main tables.

As shown in Table 25, each proxy contributes to the final performance. The necessity proxy improves discriminability by learning when relations should be rejected. The set-level sufficiency proxy improves stability across support-query splits. The distribution-level proxy further improves performance by encouraging relation principles that remain useful across multiple mini-batch environments.

E.4 Correlation between Certification Objective and Downstream Performance

The theoretical analysis introduces a proxy mismatch term between the observable certification objective and the target risk. Although this mismatch cannot be directly observed during SSL training, we can empirically examine whether the certification objective is correlated with downstream

Table 25: Ablation of observable certification proxies on ImageNet-100. The baseline and full GeSSL rows use the same setting and results as the main linear evaluation table. The additional proxies provide complementary improvements.

Method	$\mathcal{L}_{disc}$	Support-Query	Multi-Batch Env.	Top-1 ↑	Top-5 ↑
SimCLR	–	–	–	70.15 ± 0.16	89.75 ± 0.14
SimCLR + Necessity Proxy	✓	–	–	71.24 ± 0.20	90.68 ± 0.18
SimCLR + Set-level Sufficiency	–	✓	–	71.58 ± 0.22	91.05 ± 0.19
SimCLR + Distribution-level Sufficiency	–	–	✓	70.96 ± 0.21	90.43 ± 0.17
SimCLR + Necessity + Set-level Sufficiency	✓	✓	–	72.31 ± 0.23	91.76 ± 0.20
SimCLR + GeSSL	✓	✓	✓	72.96 ± 0.24	92.50 ± 0.17
MAE	–	–	–	76.56 ± 0.16	93.24 ± 0.24
MAE + Necessity Proxy	✓	–	–	77.21 ± 0.23	94.11 ± 0.22
MAE + Set-level Sufficiency	–	✓	–	77.46 ± 0.25	94.36 ± 0.24
MAE + Distribution-level Sufficiency	–	–	✓	77.04 ± 0.22	93.96 ± 0.21
MAE + Necessity + Set-level Sufficiency	✓	✓	–	77.93 ± 0.28	95.21 ± 0.23
MAE + GeSSL	✓	✓	✓	78.45 ± 0.31	96.17 ± 0.26

Table 26: Correlation between the validation certification objective and downstream linear evaluation accuracy. The positive correlations suggest that the observable certification objective is informative for downstream representation quality.

Method	Dataset	Pearson Corr. ↑	Spearman Corr. ↑
SimCLR + GeSSL	STL-10	0.78	0.82
SimCLR + GeSSL	ImageNet-100	0.74	0.79
MoCo + GeSSL	STL-10	0.76	0.80
MoCo + GeSSL	ImageNet-100	0.72	0.77
MAE + GeSSL	STL-10	0.73	0.76
MAE + GeSSL	ImageNet-100	0.70	0.74

performance. We collect checkpoints from different training epochs and hyperparameter settings, including different values of μ and k. For each checkpoint, we compute the following:

$$\mathcal{C} = \mathcal{L}_{ssl} + \mu\overline{\mathcal{L}}_{disc} + \mathcal{L}_{query}$$

on held-out relation environments, and evaluate the frozen representation with linear probing. We report Pearson and Spearman correlations between $-\mathcal{C}$ and downstream accuracy. A higher positive correlation means that a lower certification objective is associated with better downstream performance.

Table 26 shows a consistent positive correlation between the negative certification objective and downstream accuracy. The correlation is not perfect, which is expected because SSL objectives are only proxies for downstream semantic evaluation. Nevertheless, the result supports the theoretical view that improving the observable certification objective tends to improve the quality of the learned representation.

F More Discussion

F.1 Differences between GeSSL and Meta-Learning

Meta-learning (Finn et al., 2017b; Wang et al., 2024b; Snell et al., 2017), often referred to as "learning to learn", has emerged as a prominent approach to improve the efficiency and adaptability of machine learning models, especially in scenarios with limited data. The fundamental idea behind meta-learning is to train models that can rapidly adapt to new tasks with minimal data by leveraging prior experiences gained from tasks.

Few-shot Learning (Khodadadeh et al., 2019; Jang et al., 2023): One of the primary areas where meta-learning has demonstrated substantial impact is in few-shot learning. Methods like Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017b) aim to find a set of model parameters that are sensitive to changes in the task, allowing for quick adaptation to new tasks with just a few examples. Variants of MAML, such as First-Order MAML (FOMAML) and Reptile (Nichol & Schulman, 2018), reduce the computational complexity of the original algorithm while maintaining competitive performance.

Metric-based Approaches: Metric-based meta-learning methods, such as Matching Networks (Sung et al., 2018) and Prototypical Networks (Snell et al., 2017), learn an embedding space where similar tasks are closer together. These models perform classification by comparing the distance between new examples and a few labeled instances in this learned space, achieving remarkable results in few-shot classification tasks.

Memory-augmented Networks: Another line of research in meta-learning explores the use of external memory structures to facilitate rapid adaptation. Santoro et al introduced Memory-Augmented Neural Networks (MANNs) (Santoro et al., 2016) that use an external memory to store and retrieve information about past tasks, enabling the model to perform well even in tasks with highly variable distributions.

Gradient-based Meta-learning: Beyond MAML, other gradient-based methods such as Meta-SGD (Li et al., 2017) and Learning to Learn with Gradient Descent have been proposed. These methods modify the way gradients are used during the training of the model, either by learning the initial parameters or by learning the learning rates for different parameters, allowing for more efficient adaptation.

Bayesian Meta-learning: Bayesian approaches to meta-learning, such as Bayesian MAML (Zhang et al., 2021), offer a probabilistic framework for capturing uncertainty and improving generalization to new tasks. These methods have been particularly useful in scenarios where task distributions are diverse, and the model needs to account for uncertainty in task inference.

Meta-learning for Reinforcement Learning: Meta-learning has also been successfully applied in the domain of reinforcement learning (RL). Methods such as Meta-RL (Yu et al., 2020) aim to train agents that can quickly adapt to new environments by leveraging the experience gained in previous tasks. These approaches have shown promise in enabling RL agents to solve tasks with minimal exploration, a crucial aspect for real-world applications where exploration can be costly or risky.

F.2 Broader Impacts and Limitations

In this subsection, we briefly illustrate the broader impacts and limitations of this work.

Broader Impacts This work advances self-supervised learning (SSL) by introducing a counterfactual perspective to relation certification. Rather than relying on superficial similarities, GeSSL explicitly approximates counterfactual necessity and sufficiency through a unified bi-level optimization framework. This encourages the model to capture invariant semantic factors while discarding unstable nuisance variations. By providing theoretical guarantees for this certification mechanism, we formally ensure that GeSSL's learned representations will remain robust under distributional shifts and transfer effectively to unseen environments. Empirically, GeSSL delivers state-of-the-art results across diverse benchmarks, demonstrating its capability to learn highly reliable representations. Ultimately, this work introduces a novel causal-inspired lens to SSL, helping bridge the gap between empirical representation learning and causal robustness. This work benefits the field of SSL and machine learning, and also opens up exciting new avenues for future research.

Limitations While this work presents extensive empirical evidence of its effectiveness across a variety of settings, our approximation of counterfactual interventions is primarily explored within single-modality contexts, although we demonstrate performance on both images and text. We do not offer a dedicated examination of multi-modal or cross-modal scenarios (e.g., vision-language pre-training). We plan to investigate these cross-modal counterfactual relations in our future work.