INSPIRE: Understanding Local-Global Representation Learning via Controlled Self-Supervised Training and Representation Interpretability

Reem Alharthi[1], Rashid Mehmood[2]*, Aiiad Albeshri[3], Abdulaziz A. Almuzaini[4]
Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia[1, 3]
Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia[2, 4]

***Abstract*—Self-supervised learning (SSL) has become a powerful paradigm for visual representation learning, yet it remains insufficiently understood how different training strategies shape the balance between global semantic structure and local visual detail. This study introduces the INSPIRE Framework, a unified framework for systematically investigating local-global representation learning in SSL through controlled training, downstream evaluation, and representation interpretability analysis. Within this framework, we develop the INSPIRE Method, a contrastive SSL approach that constructs merged training views by combining target images with distractor content and aligning global image representations with region-level features. By varying the number of merged regions, INSPIRE explicitly controls spatial granularity during pretraining. We evaluate INSPIRE through controlled ImageNet-100 experiments (130K images) with a ResNet-18 backbone, large-scale ImageNet-1K pretraining (1.28million images) with a ResNet-50 backbone, and transfer to fine-grained recognition benchmarks. We further introduce the INSPIRE Representation Interpretability Investigation Methodology, which combines representation similarity analysis with scale-controlled spatial perturbations to examine spatial sensitivity across SSL methods and network layers. Experiments show that representation quality varies non-monotonically with spatial granularity, with intermediate granularity often producing stronger transfer performance than coarser or more fragmented alternatives. The analysis further reveals distinct spatial sensitivity profiles across SSL objectives and progressively increasing perturbation sensitivity across network depth. Runtime experiments using single-GPU and multi-GPU distributed pretraining demonstrate the practical scalability of the framework. These findings suggest that SSL representations are better understood as occupying positions along a continuum of spatial sensitivity rather than as strictly local or global learners.**

***Keywords*—Self-supervised learning; visual representation learning; local-global representation learning; representation interpretability; contrastive learning; spatial sensitivity; image representation; representation analysis; computer vision**

I. Introduction

Machine learning has achieved remarkable success across a wide range of computer vision tasks, including image classification, object detection, semantic segmentation, and visual understanding [1], [2], [3]. Much of this success has been driven by deep neural networks trained on large-scale annotated datasets. However, acquiring high-quality labels is expensive, time-consuming, and often impractical for many real-world applications. These limitations have motivated increasing interest in self-supervised learning (SSL), which seeks to learn useful visual representations directly from unlabeled data by exploiting intrinsic structure within the data itself.

Recent advances in SSL have demonstrated that representations learned without manual annotations can rival or even surpass those obtained through supervised learning on many downstream tasks [4]. Despite their success, SSL methods differ substantially in the type of visual information they encode. Effective visual representations must capture both global semantic structure and local discriminative details [5]. Global representations encode high-level information such as object identity, shape, and semantic context, whereas local representations capture fine-grained visual cues including textures, object parts, boundaries, and subtle appearance variations [6]. Different downstream tasks depend on these information sources to varying degrees. Image classification often relies heavily on global semantic understanding [7], while fine-grained recognition, object detection, and segmentation require stronger sensitivity to local visual information.

A growing body of research has attempted to improve local representation learning in SSL through region-level objectives [8], [9], patch-based learning [10], and multi-scale and hierarchical representation strategies [5], [11]. These studies demonstrate that incorporating local information can improve performance on spatially sensitive tasks. Existing research has also explored a variety of training objectives [12] and architectural designs [13] to improve the quality of learned representations. However, these approaches typically investigate multiple design factors simultaneously, making it difficult to isolate how individual training choices shape representation encodings and influence the balance between local and global information.

Another limitation concerns representation interpretability. While interpretable machine learning (IML) and explainable artificial intelligence (XAI) have traditionally focused on explaining model predictions and decision-making processes [14], [15], recent research has begun extending interpretability to learned representations [16]. This emerging direction, referred to as representation interpretability, seeks to understand

*Corresponding author.

what information is encoded within a model's internal representations, how this information is organized, and how these representations contribute to downstream behavior. Existing studies have investigated representation similarity [17], [18], semantic concepts [19], [20], and feature activations [21], [22], providing valuable insights into the structure and content of learned representations. However, they provide limited understanding of how SSL representations encode and respond to controlled spatial perturbations at different spatial scales and what these responses reveal about the balance between local and global information. Current evidence regarding spatial sensitivity is therefore inferred primarily from downstream task performance rather than measured directly at the representation level.

These observations reveal three important research gaps. First, no unified framework has been developed to systematically investigate, compare, and interpret local-global representation learning across different SSL approaches. Second, the influence of spatial granularity on representation learning remains poorly understood because it has not been investigated as an independent experimental factor separate from architectural and objective-related design choices. Third, existing representation interpretability studies do not provide a systematic methodology for investigating how SSL representations respond to controlled spatial perturbations or how these responses relate to local-global representation learning.

To address these challenges, we introduce the INSPIRE Framework, a unified framework for investigating local-global representation learning in SSL (See Section III). The framework integrates three complementary components. First, it employs controlled spatial granularity manipulation during contrastive learning, enabling the influence of spatial composition on representation learning to be investigated systematically and independently from other design factors. Second, it evaluates learned representations across both global and fine-grained downstream tasks, providing insights into how different training strategies affect transfer performance at different spatial scales. Third, it introduces a representation interpretability methodology based on scale-controlled spatial perturbations, enabling spatial sensitivity and representation behavior to be examined directly rather than inferred solely from downstream performance. These components collectively provide a structured basis for understanding how SSL training strategies shape representation encodings and local-global representation learning. Using the proposed framework, we conduct a comprehensive empirical investigation of local-global representation learning across multiple SSL configurations. The main contributions of this study are summarized as follows.

- We introduce the INSPIRE Framework, a unified framework for systematically investigating local-global representation learning in self-supervised learning through controlled training, downstream evaluation, and representation interpretability analysis.
- We develop the INSPIRE Method, a contrastive self-supervised learning approach that explicitly controls spatial granularity by constructing merged training views that combine an image with distractor content and align global image representations with region-level features extracted from these composite views. By systematically varying the number of merged regions, the method enables controlled investigation of how spatial composition and training design shape representation encodings and local-global representation learning.
- We provide a pretrained INSPIRE Encoder to support future research on local-global representation learning, spatial sensitivity, and representation interpretability in SSL.
- We introduce the INSPIRE Representation Interpretability Investigation Methodology, which combines representation similarity analysis with scale-controlled spatial perturbations to systematically investigate how SSL representations encode and respond to localized and distributed spatial information, enabling direct examination and comparison of spatial sensitivity and representation behavior across SSL methods.
- We provide new empirical insights into how SSL training strategies shape representation encodings and local-global representation learning, demonstrating non-monotonic effects of spatial granularity, distinct spatial sensitivity profiles across SSL methods, and evidence that SSL representations occupy different positions along a continuum of spatial sensitivity.

We named our framework INSPIRE to reflect its key characteristics: representation Interpretability, understanding of Neural representations, Spatial sensitivity analysis, Perturbation-based Investigation, and Representation Encodings. Beyond the specific framework and experimental findings presented in this work, the results have broader implications for the design and interpretation of self-supervised learning systems. By providing a structured approach for investigating local-global representation learning through representation interpretability and spatial sensitivity analysis, the proposed framework enables a deeper understanding of how training strategies shape learned representations. Such understanding can support the development of more effective SSL methods, guide the selection of representations for different downstream tasks, and contribute to the broader goal of making representation learning more transparent, interpretable, and scientifically grounded.

The remainder of this study is organized as follows. Section II reviews the related literature and identifies the research gaps. Section III presents the INSPIRE Framework, including the INSPIRE Method, the downstream evaluation protocol, and the INSPIRE Representation Interpretability Investigation Methodology. Section IV reports controlled ImageNet-100 experiments with a ResNet-18 backbone, examining how different merged-view patch configurations influence global and fine-grained transfer performance. Section V extends the evaluation to ImageNet-1K using a ResNet-50 backbone and compares INSPIRE with 9 well-known SSL baselines, distinguishing the controlled ablation study of Section IV from the large-scale benchmark evaluation. Section VI investigates representation interpretability through scale-controlled spatial perturbation analysis. Section VII analyzes computational cost and runtime, covering both single-GPU pretraining for

ImageNet-100 and multi-GPU distributed pretraining for ImageNet-1K. Section VIII discusses the key findings and their broader implications, and concludes the study.

II. Background and Related Work

This section reviews prior work on global and local representation learning in SSL (Sections II.A and II.B) and existing approaches to representation interpretability (Section II.C). It concludes by identifying the research gaps that motivate the proposed framework (Section II.D).

A. Global Representations in SSL Methods

Classical SSL methods were primarily developed to learn image-level representations that improve downstream tasks such as image classification. Consequently, early SSL approaches largely emphasized global semantic understanding while providing limited attention to fine-grained local structures and spatial correspondences. This section reviews the major SSL paradigms that primarily learn global semantic representations and highlights subsequent efforts to strengthen local feature learning.

Early SSL research was dominated by contrastive learning, which maximizes agreement between different augmented views of the same image while separating representations of different images in the embedding space. SimCLR [13] and MoCo [23] treat each image as a semantic instance, encouraging invariant image-level representations with strong transferability. However, their emphasis on global semantics often underrepresents fine-grained spatial relationships, motivating dense and region-aware contrastive learning methods that explicitly incorporate patch correspondence and spatial consistency [24], [25]. Beyond pairwise contrastive learning, SwAV [26] replaces explicit sample matching with online clustering and prototype assignment, enabling multiple augmented views to be associated through shared semantic clusters while remaining focused on global representations. TiCo [27] further improves transformation-invariant learning through covariance regularization that reduces feature redundancy and produces more structured global embeddings.

To remove the dependence on negative samples, non-contrastive and self-distillation methods were subsequently introduced. BYOL [12] employs an online and momentum-updated target network to learn consistent representations across augmented views, primarily reinforcing global semantic invariance. Later extensions incorporated dense and hierarchical objectives to improve local feature learning and multi-scale representations [28]. DINO [21] further extends self-distillation within Vision Transformer architectures, where patch tokenization and self-attention naturally capture both global scene semantics and localized patch-level structures, making the learned representations effective for semantic localization and fine-grained visual understanding.

Redundancy-reduction approaches, including Barlow Twins [29] and VICReg [30], learn invariant representations by decorrelating feature dimensions to avoid representational collapse while remaining primarily focused on image-level representations. A complementary direction is masked image modeling, where representations are learned by reconstructing masked image regions. MAE [31] employs an asymmetric encoder-decoder architecture to reconstruct masked patches from visible regions, directly optimizing patch-level reconstruction rather than image-level alignment. As a result, MAE naturally captures local structures, spatial relationships, and contextual information.

B. Local Representation in SSL Methods

Although classical SSL methods discussed above learn highly transferable global representations, many downstream vision tasks require fine-grained spatial information in addition to image-level semantics [5], [32]. This requirement has motivated a broad range of approaches that strengthen local representation learning through region-level objectives, patch-based modeling, hierarchical representations, and spatial regularization.

One direction extends contrastive learning to dense and region-aware settings. ReSim [15], RegionCL [8], MaskCo [9], Patch-wise SSL [10], Self-Patch [32], and HCL [24] introduce local correspondence, masked prediction, patch alignment, neighborhood consistency, and spatial feature embedding to improve localization and dense prediction. Despite their differences, these methods share the common objective of strengthening local feature learning by incorporating region-level supervision into contrastive learning.

A second direction focuses on learning hierarchical representations across multiple spatial scales. Representative approaches include Multi-level Contrastive Learning for Vision Transformers (MCVT) [33], Self-Supervised Pyramid Representation Learning (SS-PRL) [5], CsML [11], the Global-Local Self-Distillation framework [6], and UDI [34]. These methods combine representations from different network depths, feature hierarchies, or local prediction mechanisms to jointly capture coarse semantic structure and fine-grained spatial information.

A complementary line of research enhances local representations through positional supervision, spatial regularization, and masked image modeling. Positional Label [35] introduces positional prediction tasks, spatial entropy regularization [36] encourages coherent attention regions, while masked image modeling methods such as iBOT [37] and attention-guided variants such as ACoMIM [38] learn fine-grained representations through patch reconstruction combined with global semantic objectives.

C. Representation Interpretability in SSL Methods

Understanding SSL representations requires analysis beyond downstream task performance. Consequently, a growing body of research has focused on representation interpretability, investigating how learned representations are organized, what information they encode, and how they vary across architectures, training objectives, and learning paradigms. Existing studies can be broadly grouped into similarity-based analyses, probing methods, attention-based visualizations, and feature-level analyses.

Similarity-based analyses examine the relationships between representations learned by different models, network layers, and learning objectives. Among the proposed techniques, Centered Kernel Alignment (CKA) [39] has become a widely adopted

approach for comparing learned representations, offering improved robustness over earlier Canonical Correlation Analysis (CCA)-based methods [40]. Recent studies have used CKA to compare representations across architectures and training paradigms [17], [18], [41], showing that convolutional networks and Vision Transformers exhibit distinct representation organization, while supervised and self-supervised models often learn highly aligned representations in early and intermediate layers before diverging in deeper layers. These findings demonstrate that both architectural design and learning objectives substantially influence representation organization.

A complementary line of research investigates the semantic information encoded within SSL representations through probing methods [19], [20], attention-based visualizations [21], [22], and feature-level analyses [42], [43]. Probing frameworks have been used to identify the visual concepts captured by learned representations [19], [20], while attention visualization methods such as DINO [21] and ICE [22] reveal semantic object structure and region-level predictions learned without supervision. Feature-level analyses further quantify representation quality by measuring phenomena such as dimensional collapse [42].

D. Research Gap

The literature reviewed above demonstrates substantial progress in SSL representation learning and interpretation while also revealing three important research gaps. First, although prior studies have investigated global and local representation learning (Sections II.A and II.B), no unified framework has been proposed to systematically investigate, compare, and interpret local-global representation learning across SSL approaches. Second, existing methods have primarily improved downstream performance through architectural modifications [13], [23], objective design [12], [28], region-level learning [8], [9], and multi-scale representations [5], [11]. However, spatial granularity has not been systematically investigated as an independent experimental factor and remains entangled with other design choices, making its influence on local-global representation learning difficult to isolate. Third, existing representation interpretability studies have investigated representation similarity [17], [18], [41], semantic concepts [19], [20], and feature activations [21], [22] (Section II.C), but provide limited understanding of how SSL representations respond to controlled spatial perturbations at different spatial scales and how these responses relate to local-global representation learning.

These gaps motivate the INSPIRE Framework presented in Section I. The framework, its associated INSPIRE Method, INSPIRE Encoder, and INSPIRE Representation Interpretability Investigation Methodology are designed to address these limitations through a unified approach to investigating local-global representation learning. Further details are provided in Section III and onwards.

III. Methodology

The INSPIRE Framework comprises three complementary components for systematically investigating local-global representation learning in SSL (see Fig. 1). The framework integrates the INSPIRE Method and Encoder for controlled spatial granularity learning (see Section III.A), downstream evaluation across tasks requiring different levels of spatial information (Section III.B), and the INSPIRE Representation Interpretability Investigation Methodology for analyzing representation behavior through scale-controlled spatial perturbations (Section III.C).

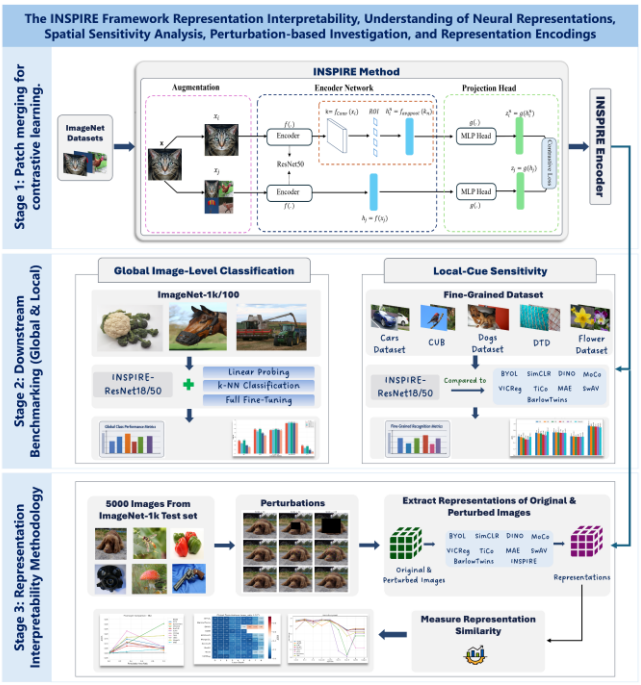


Fig. 1. INSPIRE framework: 1) Merged-view contrastive pretraining with controllable spatial granularity, 2) Evaluation on global and fine-grained tasks, and 3) Representation interpretability investigation under scale-controlled spatial perturbations.

A. INSPIRE Method

Our INSPIRE method builds upon the classic contrastive learning paradigm (e.g., SimCLR [13]) by introducing a region-based representation mechanism applied to a merged composite image. As illustrated in Fig. 2, given an input image x, two augmented views are generated. The first view, denoted x_i, is a standard augmented version of the image produced using random transformations such as cropping, flipping, or color jittering. The second view, denoted x_j, is generated by merging a transformed version of the original image x' with several other randomly selected images into a single composite image. This merged image contains multiple unrelated objects arranged in a grid-like structure, and the number of merged patches can vary (e.g., 2, 4, 6, or 8; see Fig. 2), enabling controlled manipulation of region granularity and contextual interference.

Both x_i and x_j are passed through a shared encoder network $f(\cdot)$ with a ResNet-50 backbone (Fig. 2). For the merged image x_j, we extract intermediate convolutional features after the fourth residual block, resulting in a feature map $k = f_{\text{conv}}(x_j)$. To obtain region-specific representations, we apply a Region of Interest (ROI) operation on k using bounding boxes aligned with the merged layout. Each ROI feature is then average-pooled, yielding region feature vectors $h_j^n = f_{\text{avgpool}}(k_n)$, where

n indexes the sub-image regions. Meanwhile, the standard view x_i is encoded end-to-end to produce a global representation $h_i = f(x_i)$. Next, the global representation h_i and each region vector h_j^n are passed through a shared projection head $g(.)$ implemented as an MLP, producing embeddings $z_i = g(h_i)$ and $z_j^n = g(h_j^n)$.

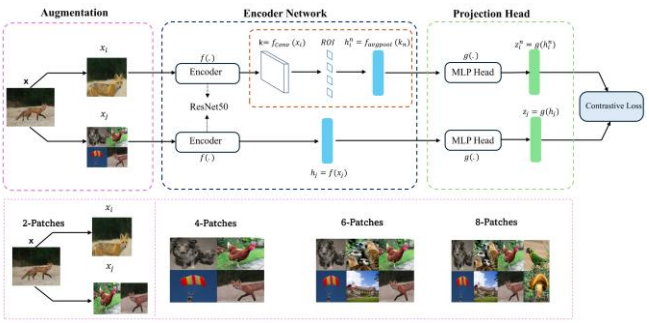


Fig. 2. INSPIRE method: global–region contrastive learning using merged views with controllable patch granularity and contextual interference.

To train INSPIRE, we adopt the normalized temperature-scaled cross-entropy (NT-Xent) objective [44] used in SimCLR, with a single positive per anchor. For each anchor embedding z_i (from x_i), we define exactly one positive z_j^+, which corresponds to the ROI embedding extracted from the merged view x_j that contains the transformed instance x'. All other embeddings, including region embeddings from non- matching ROIs and embeddings from other images in the minibatch, are treated as negatives. The contrastive loss is defined in Eq. (1), where sim( ) denotes cosine similarity between L_2-normalized embeddings, τ is a temperature hyperparameter, and M is the total number of candidate embeddings included in the contrastive normalization pool for the minibatch. This objective encourages the global representation of x_i to align with the correct ROI representation within the merged view, despite the distracting context introduced by the other merged patches.

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\mathrm{sim}(z_i, z_j^+)/\tau)}{\sum_{k=1}^{M} \mathbf{1}_{[k \neq i]} \exp(\mathrm{sim}(z_i, z_k)/\tau)}, \qquad (1)$$

B. Evaluation Protocol for Global and Local Representations

The second component of the INSPIRE Framework examines the characteristics of the representations learned by the INSPIRE Method through downstream recognition tasks. Specifically, it evaluates these representations using settings that differ in their reliance on holistic semantic information and fine-grained local visual cues, enabling both comparison with established SSL methods and assessment of local-global representation learning. We organize this evaluation into two complementary groups. For global semantic transfer, we use standard image-level classification and report performance under three protocols: k -NN classification in the learned embedding space, linear probing with a frozen encoder and a trained linear classifier, and full fine-tuning of the encoder and classifier. For local-cue sensitivity, we transfer the same pretrained encoders to fine-grained recognition benchmarks, where discriminative performance typically depends on subtle texture, part-level, and shape cues, and therefore provides a stronger test of locality-sensitive representations. Table I summarizes the datasets and train/validation split sizes.

TABLE I. DATASETS USED FOR GLOBAL CLASSIFICATION AND FINE-GRAINED (LOCAL-CUE) TRANSFER

Dataset	Representation Focus	Classes	Train	Val
ImageNet-100 [45]	Global (image-level)	100	130,000	5,000
ImageNet-1k [45]	Global (image-level)	1,000	1,281,167	50,000
Flowers-102 [46]	Local-cue sensitivity (fine-grained)	102	6,552	818
Stanford Dogs [47]	Local-cue sensitivity (fine-grained)	120	16,465	4,115
DTD [48]	Local-cue sensitivity (fine-grained)	47	4,513	1,127
CUB-200-2011 [49]	Local-cue sensitivity (fine-grained)	200	5,994	5,794
Stanford Cars [50]	Local-cue sensitivity (fine-grained)	196	8,144	8,041

We first conducted a controlled ablation study using a ResNet-18 backbone pretrained on ImageNet-100. In this stage, we isolate the effect of region granularity by varying only the number of tiles $P \in \{2,4,6,8\}$ (Fig. 2) used to construct the merged view, while keeping all other pretraining components fixed (training schedule, batch size, data augmentations, optimizer, temperature, and projection head). We denote the resulting variants as INSPIRE-P2, INSPIRE-P4, INSPIRE-P6, and INSPIRE-P8. After pretraining, we evaluate each variant on global image-level classification using the same three protocols (k-NN, linear probe, and fine-tuning) to quantify how the patch configuration affects global transfer. To evaluate local-cue sensitivity, we transfer the pretrained encoders to five fine-grained and texture recognition benchmarks (Table I) using either linear probing or partial adaptation, depending on dataset difficulty. These evaluations test whether varying P shifts representations toward global separability or toward preserving fine local discriminative cues.

For large-scale pretraining, we adopt a two-stage design. Guided by the ResNet-18 ablation, we select the patch configuration P that yields the strongest overall behavior in our global-versus-local evaluation suite and use this configuration to pretrain a ResNet-50 INSPIRE model on ImageNet-1k [45]. We then compare this ImageNet-1k pretrained INSPIRE model against established SSL baselines (e.g., SimCLR, MoCo, BYOL) under the same evaluation protocols. This setup provides a consistent and reproducible methodology for analyzing how the INSPIRE pretraining signal, controlled through the patch configuration P, impacts the balance between global semantic structure and locally discriminative feature learning.

C. INSPIRE Representation Interpretability Methodology

The third component of the INSPIRE Framework examines the spatial sensitivity of learned representations through a representation interpretability methodology based on scale-controlled spatial perturbations. By measuring how representation similarity changes in response to perturbations applied at different spatial scales, this component characterizes whether SSL representations are more sensitive to localized visual information or to broader spatial structure.

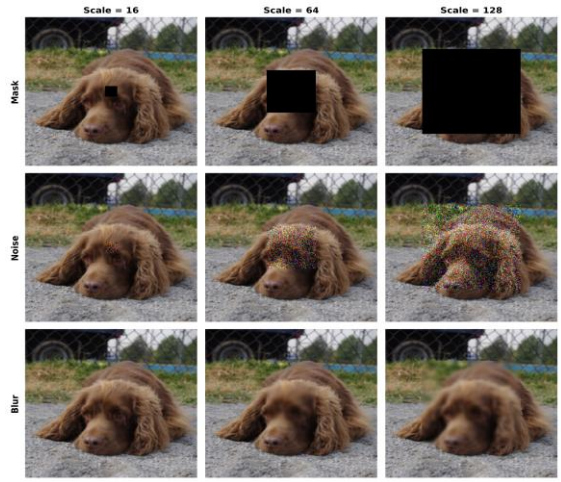


Fig. 3. Scale-controlled spatial perturbations (masking, noise, blur) applied to regions of increasing size while preserving alignment, enabling analysis of representation sensitivity to perturbation scale.

Let $x \in \mathbb{R}^{3\times H\times W}$ denote an input image and let $f(\cdot)$ represent the feature extraction function of a model at a given layer. We construct a perturbed version of the image, denoted $T_s(x)$, where T is a perturbation operator and s controls its spatial extent. We consider perturbations that preserve spatial alignment, including patch masking, localized Gaussian noise, and localized Gaussian blur. Each perturbation is applied to a square region of size $s \times s$ centered in the image, ensuring that spatial correspondence between the original and perturbed inputs is maintained (Fig. 3). By varying s, we obtain perturbations ranging from highly localized modifications to transformations affecting the entire image. To provide a scale-invariant measure, we define the normalized perturbation area in Eq. (2) as the fraction of the image affected by the perturbation.

$$\alpha = \frac{s^2}{H\times W}, \tag{2}$$

For each image x and its perturbed counterpart $T_s(x)$, we extract intermediate representations $f(x)$ and $f(T_s(x))$ from multiple layers of the model. To preserve spatial structure while obtaining fixed-dimensional representations, we apply grid-based pooling. Convolutional feature maps of shape $[B, C, H, W]$ are adaptively pooled to a fixed grid of size $g \times g$, and then flattened into vectors in $\mathbb{R}^{Cg^2}$. For transformer-based architectures, token embeddings are reshaped into spatial grids when possible, ensuring consistency across model families. This representation design retains coarse spatial layout, allowing localized perturbations to influence the resulting features.

To quantify similarity between representations, we use linear CKA [39]. Given two representations $X = f(x)$ and $Y = f(T_s(x))$, CKA is defined as:

$$\mathrm{CKA}(X,Y) = \frac{\|Y^\top X\|_F^2}{\|X^\top X\|_F \, \|Y^\top Y\|_F} \tag{3}$$

For each model, layer, perturbation type, and spatial scale s, we compute the representation change for each image x in the dataset as:

$$\Delta\mathrm{CKA}(s) = 1 - \mathrm{CKA}(f(x), f(T_s(x))) \tag{4}$$

We use a subset of 5,000 images sampled from the ImageNet-1K test set and report the mean $\Delta\mathrm{CKA}(s)$ across all samples. The resulting measurements define a scale-dependent sensitivity profile for each model and layer. By analyzing how $\Delta\mathrm{CKA}(s)$ varies as a function of the perturbation scale s, or equivalently the normalized perturbation area $\alpha = s^2/H \times W$, we obtain a structured characterization of how representations respond to perturbations at different spatial extents.

Localized perturbations provide a natural probe of spatial information in deep representations because they selectively disrupt restricted image regions while preserving the rest of the input. This approach is closely related to occlusion-based analyses introduced by Zeiler and Fergus [51], which assess the contribution of local regions by measuring changes in model responses under masking. It also connects to studies of texture–shape bias, which contrast reliance on local texture cues with more global shape information [52], [53]. In addition, the use of CKA is grounded in prior work demonstrating its effectiveness as a similarity measure for neural representations across models and layers (See Section II.C).

This component of our framework enables a controlled analysis of how representations change as a function of perturbation scale, providing a systematic way to examine sensitivity to localized and distributed modifications. By measuring representation change across layers and spatial extents, we obtain insight into how different models respond to perturbations of varying scale. While this analysis does not directly measure reliance on specific types of visual information, the observed sensitivity patterns can provide indicative evidence of how representations encode and preserve information across spatial scales.

IV. Results: ImageNet-100 (ResNet-18) (130K Images)

We analyze here INSPIRE under a controlled setting using ImageNet-100 pretraining with a ResNet-18 backbone, focusing on how the merged-view patch configuration (P) influences both global and local aspects of the learned representations while keeping all other training components fixed. Section IV.A evaluates global image-level representation quality using kNN, linear probing, and end-to-end fine-tuning, reflecting the ability to capture global semantic structure. Section IV.B examines local-cue sensitivity through transfer to fine-grained recognition benchmarks that rely on localized visual details (Table II).

TABLE II. INSPIRE Pretraining Configuration on ImageNet-100 with a ResNet-18 Backbone

Setting	Value
Backbone / Data	ResNet-18 / ImageNet-100
Training budget	100 epochs, batch size 256
Optimizer	LARS (LR 0.3, momentum 0.9, WD 10^{-6})
LR schedule	Cosine with warmup (warmup = 10 × steps/epoch)
Loss	NT-Xent (InfoNCE), $\tau = 0.4$
Projection head	SimCLR-style MLP, output dim 128
Ablation variable	Patch count $P \in \{2,4,6,8\}$
INSPIRE positives/negatives	One positive ROI (tile containing x'); others are negatives
ROI feature extraction	After layer4; global average pooling

A. Global Image-Level Results

For the ImageNet-100 experiments, we pretrain INSPIRE with a ResNet-18 backbone under a single, fixed pretraining configuration. Across all ImageNet-100 runs, the training schedule, optimizer, augmentation pipeline, projection head, and contrastive objective are held constant, and the only factor varied is the merged-view patch count $P \in \{2, 4, 6, 8\}$. INSPIRE pretraining settings are summarized in Table IV; all models are pretrained for 100 epochs with NT-Xent and a SimCLR-style MLP projection head under a fixed optimization schedule.

For comparison, we train a SimCLR baseline with the same ResNet-18 backbone (SimCLR-R18) under an identical pretraining configuration on ImageNet-100. The SimCLR model is implemented using the Lightly framework [54], and all components, including the augmentation pipeline, optimizer, learning rate schedule, batch size, number of epochs, projection head architecture, and NT-Xent loss parameters, are kept the same as in INSPIRE. This controlled setup ensures that any observed differences are attributable to the merged-view design and region-based alignment in INSPIRE. Additionally, we assess the global image-level performance of the two methods using three standard protocols (kNN evaluation, linear probing, and end-to-end fine-tuning), following the settings summarized in Table III.

TABLE III. GLOBAL EVALUATION SETTINGS

Protocol	Description
kNN	Neighbors: $k = 200$. Similarity scaling: temperature $= 0.1$. Preprocessing (train/val): resize to 256, center crop to 224, then ImageNet normalization. Reported metrics: Top-1 and Top-5 validation accuracy.
Linear probe	Trainable parameters: linear classifier only (encoder fully frozen). Training length: 90 epochs. Optimizer: SGD with learning rate 0.1, momentum 0.9, weight decay 0. Schedule: cosine learning-rate decay with no warmup. Reported metric: Top-1 validation accuracy.
Fine-tuning	Trainable parameters: encoder and classifier optimized end-to-end. Training length: 30 epochs. Optimizer: SGD with momentum 0.9, weight decay 0. Learning rate: base LR $0.05 \cdot \text{batch} \cdot \text{world}/256$. Schedule: cosine decay with no warmup. Augmentations: training uses RandomResizedCrop(224) and horizontal flip; validation uses resize to 256 and center crop to 224 (ImageNet normalization). Reported metrics: Top-1 and Top-5 validation accuracy.

Fig. 4 presents the ImageNet-100 results after 100-epoch pretraining for SimCLR-R18 and INSPIRE with different patch counts $P \in \{2, 4, 6, 8\}$. The x-axis enumerates the methods, while the y-axis reports accuracy under several evaluation protocols, including linear probing (Top-1 and Top-5), fine-tuning (Top-1 and Top-5), and kNN. A consistent pattern is that global image-level performance varies with the patch configuration, and the differences are most visible for protocols that rely directly on the pretrained representation without extensive end-to-end adaptation (kNN and linear probing). This behavior is plausible given that changing P simultaneously changes region granularity and the amount of competing context in the merged view, which may affect how easily global, class-separable structure emerges in the frozen feature space.

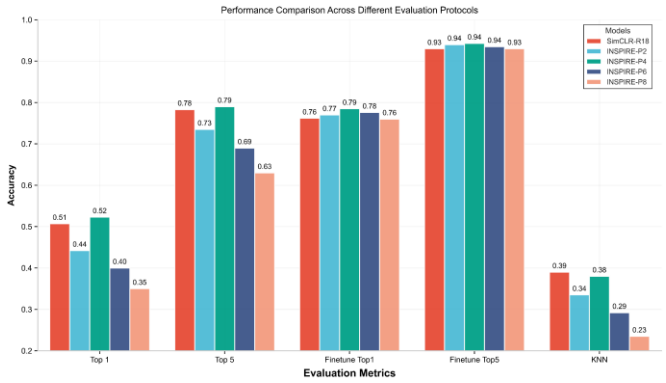


Fig. 4. ImageNet-100 results after 100-epoch pretraining: SimCLR-R18 vs. INSPIRE (P2–P8) under different evaluation protocols.

For linear probing, SimCLR-R18 reaches 0.51 Top-1 and 0.78 Top-5, while INSPIRE-P4 reaches 0.52 Top-1 and 0.79 Top-5, indicating a small improvement for P4. In comparison, the remaining INSPIRE settings show lower linear-probe accuracy, with P2 at 0.44/0.74, P6 at 0.40/0.69, and P8 at 0.35/0.63 (Top-1/Top-5). One possible interpretation is that moderate merging (P4) introduces additional contextual variation that can encourage robustness while still preserving enough global semantic consistency to support class separation under a frozen encoder. By contrast, very small (P2) or heavier merging (P6–P8) may introduce either insufficient diversity or stronger interference, which could make the alignment objective harder and reduce linear separability when only a linear head is trained.

The kNN evaluation shows a similar trend and provides an additional view of the embedding structure. SimCLR achieves 0.39 kNN accuracy, and INSPIRE-P4 is close at 0.38, whereas P2, P6, and P8 decrease to 0.34, 0.29, and 0.24, respectively. Since kNN depends on neighborhood relationships in the learned feature space, the lower kNN scores at larger P are consistent with representations that are less globally clustered by class under the frozen encoder, potentially due to increased competing content in the merged view.

After fine-tuning, the performance differences across methods become smaller, suggesting that end-to-end adaptation can partially mitigate the gaps observed under frozen-feature evaluations. SimCLR-R18 attains 0.76 Top-1 and 0.93 Top-5, while INSPIRE-P4 reaches 0.79 Top-1 and 0.94 Top-5, reflecting a modest improvement. The other INSPIRE variants cluster in a narrower range after fine-tuning (Top-1 around 0.76–0.78), even though their linear probing and kNN results are lower. This pattern is consistent with the idea that fine-tuning can adjust earlier features and decision boundaries to better match the supervised objective, reducing the sensitivity to differences in the initial embedding geometry induced by the patch configuration. Fig. 4 suggests that P controls a trade-off between region granularity and contextual interference, which is most apparent when evaluating the frozen representation space and becomes less pronounced after end-to-end adaptation.

TABLE IV. OVERVIEW OF BENCHMARK DATASETS FOR FINE-GRAINED VISUAL RECOGNITION

Dataset	Examples			Why It Is Fine-Grained
Flowers-102 [46]				Intra-class variation is low while inter-class differences are subtle, relying on fine details such as petal structure, color gradients, and reproductive parts.
	Pink Primrose	Toad Lily	Tiger Lily	
Stanford Dogs [47]				High visual similarity across breeds with distinctions concentrated in subtle facial features, fur texture, and body proportions.
	Siberian Husky	Alaskan Malamute	Samoyed	
DTD [48]				Categories are defined by abstract texture attributes rather than objects, requiring recognition of local repetitive patterns and statistical appearance cues.
	Banded	Braided	Bubbly	
CUB-200-2011 [49]				Fine-grained recognition depends on small differences in part configurations, plumage patterns, and coloration, often localized to specific body regions.
	Red-cockaded Woodpecker	Downy Woodpecker	Louisiana Waterthrush	
Stanford Cars [50]				Subtle inter-class variations arise from minor differences in shape, grille design, headlights, and brand-specific styling cues across visually similar vehicles.
	Audi A5 Coupe 2012	BMW 4 Coupe 2012	Acura TSX Sedan 2012	

B. *Fine-Grained Transfer (Local-Cue Evaluation)*

To examine local-cue sensitivity, we evaluate whether representations learned by INSPIRE transfer to fine-grained recognition tasks that require discriminative texture- and part-level cues. We compare INSPIRE variants against SimCLR-R18 trained with different merged-view patch counts. All models are pretrained on ImageNet-100 for 100 epochs under the same setup, differing only in P, and then transferred to fine-grained benchmarks to evaluate how patch count affects performance on localized visual tasks.

Table IV summarizes the fine-grained transfer datasets used to evaluate local-cue sensitivity. These benchmarks are challenging because many classes share strong global similarity, and correct recognition often depends on localized cues such as texture primitives (DTD), part-level attributes and markings (CUB-200-2011), and small shape or appearance differences within a common object category (Flowers-102, Stanford Dogs, and Stanford Cars). As a result, performance on these datasets is less driven by coarse, global semantics and more influenced by the quality of features that preserve discriminative local information.

To ensure a fair and consistent transfer evaluation, we use a fixed tuning budget for all methods. For each fine-grained dataset, we run a small set of four training configurations and apply the same set to every pretrained encoder. All transfer runs are trained for a fixed 20 epochs, and the final reported score for each (model, dataset) pair is the best validation Top-1 accuracy among these four configurations. For the linear-probe datasets (Flowers-102, Stanford Dogs, and DTD), the four configurations cover two weight decay settings (0 and 10^{-4}) and two label-smoothing values (0.05 and 0.10), while keeping the learning rate fixed at 0.3. For the more challenging datasets (CUB-200-2011 and Stanford Cars), we adopt the layer4-adaptation protocol and tune weight decay over $\{10^{-4}, 10^{-3}\}$ with label smoothing set to 0, again using a learning rate of 0.3. This procedure keeps the tuning effort comparable across methods while reducing sensitivity to a single choice of optimization hyperparameters.

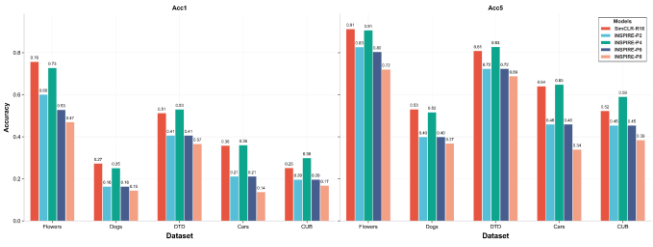


Fig. 5. Performance comparison across fine-grained transfer datasets.

Fig. 5 summarizes fine-grained transfer performance across five datasets. The x-axis lists the datasets, and the y-axis reports accuracy. The left panel shows Top-1 accuracy (Acc1), and the

right panel shows Top-5 accuracy (Acc5) for SimCLR-R18 and INSPIRE with $P \in \{2, 4, 6, 8\}$. The figure indicates that transfer performance depends on the patch configuration, with the clearest differences appearing in the more challenging benchmarks (Cars and CUB) and in the texture-focused benchmark (DTD). This behavior is consistent with the idea that increasing P changes both the effective region granularity and the amount of competing context present during pretraining, which may influence how well locally discriminative cues are preserved for downstream transfer.

For the linear-probe datasets (Flowers, Dogs, and DTD), SimCLR-R18 achieves 0.76/0.91 (Top-1/Top-5) on Flowers, 0.27/0.53 on Dogs, and 0.51/0.81 on DTD. INSPIRE-P4 reaches 0.73/0.91 on Flowers and 0.25/0.52 on Dogs, which remains close to SimCLR, while on DTD it reaches 0.53/0.83, showing a modest improvement over SimCLR on the texture-oriented task. In contrast, P2 yields lower accuracy across these datasets (e.g., 0.60/0.83 on Flowers and 0.41/0.72 on DTD), and performance decreases further for larger patch counts (P6 and P8), which may reflect increased contextual interference in the merged view when the number of tiles becomes larger.

For the layer 4-adaptation datasets (Cars and CUB), the same trend is visible, and differences become more pronounced. On Cars, SimCLR-R18 reaches 0.36/0.64, while INSPIRE-P4 reaches 0.36/0.65, indicating similar Top-1 and a small increase in Top-5. On CUB, SimCLR-R18 reaches 0.25/0.52, whereas INSPIRE-P4 reaches 0.30/0.59, reflecting a clearer improvement under the limited adaptation setting. Meanwhile, higher patch counts again show lower transfer accuracy (e.g., P8 reaches 0.14/0.34 on Cars and 0.17/0.39 on CUB), which is consistent with the possibility that stronger merging makes it harder for the pretrained representation to support fine-grained distinctions under constrained supervision.

Fig. 5 suggests that moderate merging (here, P4) can maintain competitive transfer on several fine-grained benchmarks and may provide modest gains on datasets that rely strongly on local patterns (DTD) or part-level distinctions (CUB), while heavier merging configurations (P6–P8) tend to reduce fine-grained transfer performance.

V. Results: ImageNet-1k (ResNet-50) (1.28M Images)

Having identified the best-performing patch configuration on ImageNet-100 in Section IV, we now extend the analysis to a large-scale setting by pretraining INSPIRE on ImageNet-1k with a ResNet-50 backbone and comparing it against 9 well-known self-supervised learning baselines. The goal here is to examine whether the observed effects of the merged-view design persist at scale and under more challenging evaluation conditions.

We evaluate representations along two complementary dimensions. First, in Section V.A, we assess global image-level performance using standard ImageNet protocols, including kNN, linear probing, and end-to-end fine-tuning, which reflect global semantic separability and transfer efficiency. Second, in Section V.B, we evaluate fine-grained transfer performance on datasets that depend more heavily on localized visual cues. These evaluations allow us to assess how INSPIRE behaves at scale and whether it maintains a balance between global semantic structure and locally discriminative features when compared to existing SSL methods.

A. Global Evaluation vs Nine SSL Baselines

For the large-scale global evaluation, we pretrain INSPIRE on ImageNet-1k using a ResNet-50 backbone and the $P = 4$ patch configuration (INSPIRE-P4). Training is conducted for 100 epochs using distributed data parallel (DDP) over 7 GPUs with a per-device batch size of 128 (effective batch size 896) under mixed-precision. The effective batch size was selected to balance training efficiency with the available computational resources, as substantially larger batch sizes would require additional GPU memory and computational capacity. We adopt SimCLR-style data augmentations and optimize the NT-Xent contrastive objective with temperature $\tau = 0.4$. Optimization uses LARS with momentum 0.9 and weight decay 10^{-6}. The learning rate is scaled with the effective batch size as $\text{LR} = 0.075 \cdot \sqrt{128 \times 7}$, and we apply a cosine warmup schedule with warmup length set to $10 \times$ steps-per-epoch. We use the same SimCLR-style projection head as in the ImageNet-100 experiments, mapping features to a 128-dimensional embedding space.

We compare INSPIRE-R50 against nine widely adopted SSL baselines spanning the major paradigms of modern SSL, including contrastive, clustering-based, redundancy-reduction, self-distillation, and masked image modeling methods. All methods are evaluated using the same global evaluation protocols described in Section IV.A. All evaluation settings (including kNN parameters, preprocessing, and optimization hyperparameters for linear probing and fine-tuning) are kept identical to the ImageNet-100 protocol to enable a consistent large-scale comparison focused on representation quality.

All large-scale comparisons in this section follow a consistent benchmarking setup based on the Lightly framework [54]. The SSL baselines (e.g., DINO, SwAV, SimCLR, BYOL, MoCo v2, VICReg, Barlow Twins, TiCo, and MAE) are taken from Lightly-pretrained checkpoints trained for 100 epochs with an effective batch size of 256 and evaluated under the same protocol suite used throughout this work.

Fig. 6. ImageNet-1k linear probe results (ResNet-50, 100 epochs): Top-1 and Top-5 accuracy (%) for INSPIRE-R50 and SSL baselines.

Fig. 6 summarizes the ImageNet-1k linear evaluation results after 100 epochs of pretraining with a ResNet-50 backbone. The

x-axis lists the SSL methods, and the y-axis reports linear-probe accuracy (Top-1 and Top-5, in %), obtained under the same linear evaluation protocol.

INSPIRE-R50 achieves 65.4 Top-1 and 87.2 Top-5. Among the methods compared, DINO (68.2 Top-1, 87.9 Top-5) and SwAV (67.2 Top-1, 88.1 Top-5) obtain the highest Top-1 accuracy in this setting, while INSPIRE-R50 is lower in Top-1 and closes in Top-5. Relative to several commonly used contrastive and redundancy-reduction baselines, INSPIRE-R50 shows higher linear-probe accuracy than SimCLR (63.2/85.2), VICReg (63.0/85.4), Barlow Twins (62.9/84.3), BYOL (62.5/85.0), and MoCo v2 (61.5/84.1) in both Top-1 and Top-5. TiCo (49.7/74.4) and MAE (46.0/70.2) are lower under this linear evaluation protocol. Fig. 6 suggests overall that INSPIRE-R50 yields a competitive linear evaluation performance on ImageNet-1k relative to several established SSL baselines, while remaining below the best-performing methods in this comparison.

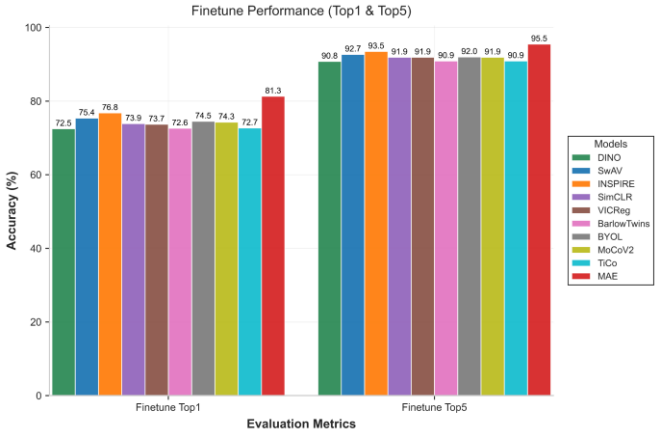


Fig. 7. ImageNet-1k fine-tuning results (ResNet-50, 100 epochs): Top-1 and Top-5 accuracy (%) for INSPIRE-R50 and SSL baselines.

Fig. 7 shows the ImageNet-1k fine-tuning results after 100 epochs of ResNet-50 pretraining. The x-axis lists the SSL methods, and the y-axis reports fine-tuned accuracy (Top-1 and Top-5, in %), obtained under the same end-to-end fine-tuning protocol.

INSPIRE-R50 reaches 76.8 Top-1 and 93.5 Top-5. In this setting, INSPIRE-R50 is above several commonly used SSL baselines, including SimCLR (73.9/91.9), VICReg (73.7/91.9), BYOL (74.5/92.0), MoCo v2 (74.3/91.9), Barlow Twins (72.6/90.9), DINO (72.5/90.8), and TiCo (72.7/90.9), and it is also slightly above SwAV (75.4/92.7). MAE shows a different behavior in this comparison: it attains the highest fine-tuning accuracy (81.3 Top-1 and 95.5 Top-5), despite being much lower in the linear-probe results reported earlier. This contrast is consistent with the notion that some pretraining objectives may produce features that are not as linearly separable without adaptation yet become highly effective once fine-tuned with supervised labels. In this context, INSPIRE Method's results appear relatively stable across evaluation protocols, providing competitive linear evaluation performance while also achieving strong accuracy after end-to-end fine-tuning.

Fig. 8 reports the kNN evaluation results on ImageNet-1k after 100-epoch pretraining with a ResNet-50 backbone. The x-axis lists the SSL methods, and the y-axis shows kNN accuracy (Top-1 and Top-5, in %), computed under the same kNN protocol used throughout this section.

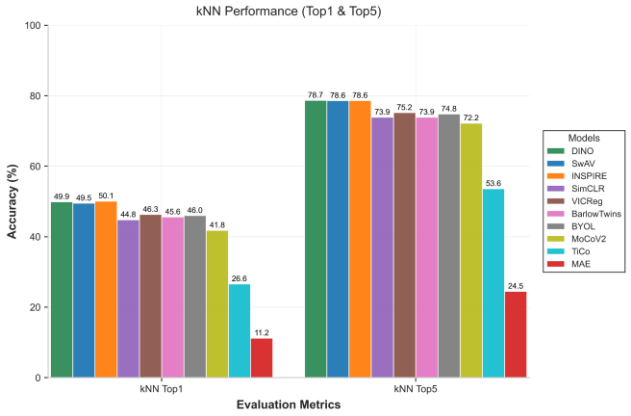


Fig. 8. ImageNet-1k kNN results (ResNet-50, 100 epochs): Top-1 and Top-5 accuracy (%) for INSPIRE-R50 and SSL baselines.

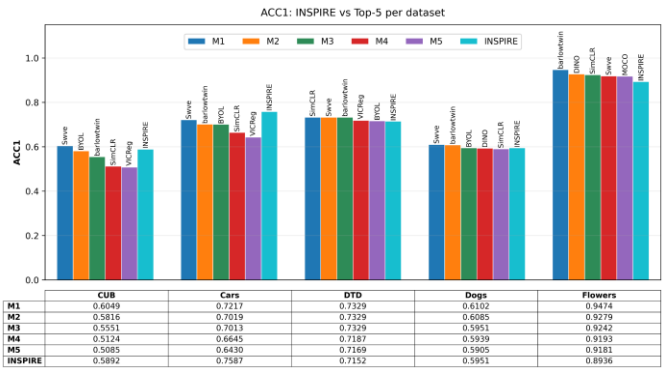
INSPIRE-R50 achieves 50.1 Top-1 and 78.6 Top-5. In this comparison, INSPIRE-R50 is close to the strongest kNN results reported by DINO (49.9/78.7) and SwAV (49.5/78.6), and it is higher than several commonly used baselines such as SimCLR (44.8/73.9), VICReg (46.3/75.2), BYOL (46.0/74.8), Barlow Twins (45.6/73.9), and MoCo v2 (41.8/72.2). TiCo (26.6/53.6) and MAE (11.2/24.5) are lower under kNN, which is consistent with the general observation that kNN strongly depends on the class structure being directly reflected in the embedding space without supervised adaptation.

Across the three global protocols reported in this section (linear probing in Fig. 6, fine-tuning in Fig. 7 , and kNN in Fig. 8), INSPIRE-R50 shows consistent performance under both frozen-feature evaluations (kNN and linear probing) and end-to-end adaptation (fine-tuning). While the relative ordering differs across protocols for some methods (e.g., MAE performs substantially better after fine-tuning than under kNN or linear probing), INSPIRE-R50 remains broadly competitive across the full evaluation suite, suggesting its suitability as a stable large-scale pretraining approach in this setting.

B. Fine-Grained Transfer vs Nine SSL Baselines

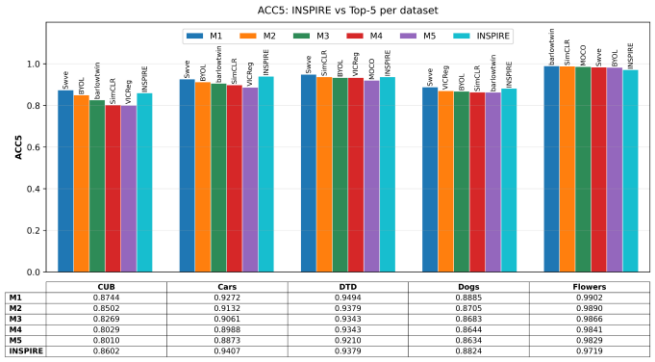
Here, we extend the large-scale evaluation by comparing fine-grained transfer performance of INSPIRE-R50 against a set of established SSL baselines, including BYOL, DINO, MAE, MoCo, SimCLR, SwAV, TiCo, and VICReg. The goal is to assess how representations learned at ImageNet-1k scale transfer to recognition problems that typically depend on subtle, localized cues such as fine textures, part-level attributes, and small shape variations.

To ensure a fair comparison across methods, we follow the same fine-grained evaluation protocol introduced in Section IV.B All pretrained models are evaluated on the same set of fine-grained datasets using identical transfer settings and a shared hyperparameter configuration.



	CUB	Cars	DTD	Dogs	Flowers
M1	0.6049	0.7217	0.7329	0.6102	0.9474
M2	0.5816	0.7019	0.7329	0.6085	0.9279
M3	0.5551	0.7013	0.7329	0.5951	0.9242
M4	0.5124	0.6645	0.7187	0.5939	0.9193
M5	0.5085	0.6430	0.7169	0.5905	0.9181
INSPIRE	0.5892	0.7587	0.7152	0.5951	0.8936

Fig. 9. Fine-grained transfer Top-1 (ACC1): INSPIRE-R50 vs best SSL baselines per dataset.



	CUB	Cars	DTD	Dogs	Flowers
M1	0.8744	0.9272	0.9494	0.8885	0.9902
M2	0.8502	0.9132	0.9379	0.8705	0.9890
M3	0.8269	0.9061	0.9343	0.8683	0.9866
M4	0.8029	0.8988	0.9343	0.8644	0.9841
M5	0.8010	0.8873	0.9210	0.8634	0.9829
INSPIRE	0.8602	0.9407	0.9379	0.8824	0.9719

Fig. 10. Fine-grained transfer Top-5 (ACC5): INSPIRE-R50 vs best SSL baselines per dataset.

Fig. 9 and Fig. 10 summarize the fine-grained transfer comparison between INSPIRE-R50 and a subset of SSL baselines on five datasets. In both figures, the x-axis lists the datasets (CUB, Cars, DTD, Dogs, and Flowers) and the y-axis reports accuracy. Fig. 9 shows Top-1 (ACC1) and Fig. 10 shows Top-5 (ACC5). For readability, the figures display INSPIRE together with the top-performing baseline methods per dataset (the specific baseline names are indicated above the bars), so the compared baseline set can differ slightly from one dataset to another.

Fig. 9 shows that INSPIRE achieves comparable Top-1 performance across all fine-grained datasets. On Cars, INSPIRE reaches 75.87%, which is higher than the other reported methods (≈70–72%). On CUB, INSPIRE (58.92%) is slightly below the best baseline, SwAV (60.49%), but remains within a close range. A similar pattern appears on Dogs, where INSPIRE (59.51%) is near SwAV (61.02%) and other baselines. On DTD, INSPIRE (71.52%) is below SimCLR and SwAV (≈73.29%), and on Flowers (89.36%) it is lower than methods such as Barlow Twins (94.74%) and DINO (≈93%). The INSPIRE Method overall provides consistent performance across datasets, although it does not achieve the highest accuracy in all cases.

Fig. 10 (ACC5) follows a similar pattern, where INSPIRE maintains comparable performance across all datasets. On Cars, INSPIRE achieves 94.07%, which is above other methods such as SwAV (92%) and BYOL (91.32%). On CUB, INSPIRE (86.02%) is slightly lower than SwAV (87.44%). On Dogs, INSPIRE (88.24%) is comparable to SwAV (88.85%) and other baselines. On DTD, INSPIRE (93.79%) is close to SwAV (94.94%), while on Flowers (97.19%) it trails Barlow Twins (99.02%) and SimCLR (≈98.9%). Compared to Top-1 results, the differences between methods are smaller in Top-5 accuracy, indicating that INSPIRE captures relevant class information even when it is not ranked highest in Top-1 accuracy.

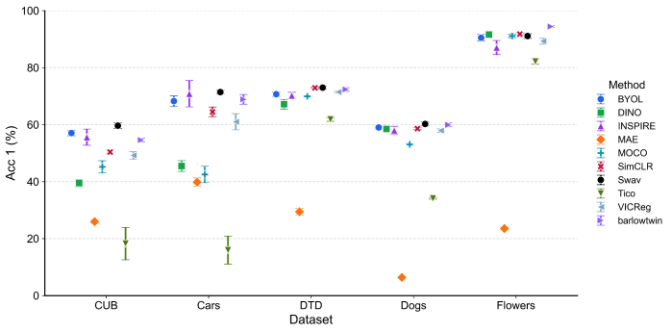


Fig. 11. Mean and standard deviation of the Top-1 validation accuracy across 4 randomized runs per dataset.

Note that, for the experiments presented in this section, we performed four independent runs for each method on each dataset. Fig. 11 reports the mean and standard deviation of the Top-1 validation accuracy to illustrate the consistency and stability of the results. All results presented throughout the study are reported as mean performance across multiple independent executions. To maintain the focus and readability of the study, we do not further analyze these statistics. Future work may investigate the variability and robustness of different SSL methods, including the proposed INSPIRE Method, in greater detail.

VI. Results: Representation Interpretability

While the previous sections (Sections IV and V) evaluate SSL methods through downstream performance, this section adopts a representation-level interpretability perspective to directly investigate the behavior of learned representations. Building on the methodology of our framework introduced in Section III.C, we investigate how representations learned by different SSL methods respond to controlled perturbations across network layers and spatial scales. This analysis complements downstream evaluation by characterizing representation sensitivity and robustness under varying perturbation conditions, providing insight into how different SSL objectives may influence representation behavior.

The top panel of Fig. 12 presents the average CKA similarity between representations learned by different SSL methods, aggregated across all layers. The results show moderate similarity across methods (≈0.30–0.52), indicating that while SSL approaches share common representational structure, they also exhibit method-specific differences. Methods based on invariance learning, including both contrastive and redundancy-reduction approaches such as SimCLR, VICReg, and Barlow Twins, exhibit the highest mutual similarity (e.g., VICReg–SimCLR: 0.52, SimCLR–Barlow Twins: 0.51, VICReg–Barlow Twins: 0.50), suggesting that despite differences in their objectives, they produce similar geometric constraints on the representation space.

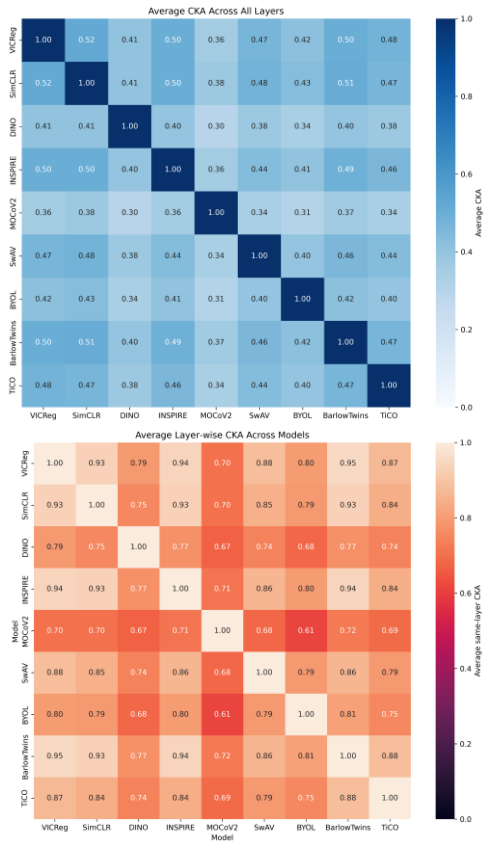


Fig. 12. Average CKA across SSL models. The top panel shows average CKA across all layers, while the bottom panel shows average layer-wise CKA between corresponding layers.

In contrast, DINO consistently shows lower similarity with other methods (≈0.38–0.41 with most approaches), reflecting its distinct self-distillation objective, which encourages clustering-like structure and semantic grouping rather than strict instance-level discrimination. MoCoV2 also exhibits relatively lower similarity (≈0.30–0.38), suggesting that differences in training mechanisms, such as the use of momentum encoders and memory queues, may contribute to distinct representation structures. Notably, INSPIRE shows strong alignment with invariance-based methods (e.g., 0.50 with SimCLR and VICReg, 0.49 with Barlow Twins) while maintaining measurable differences from other approaches (≈0.40 with DINO and 0.36 with MoCoV2), suggesting that it preserves core contrastive representation properties while introducing controlled variation in spatial representation learning.

The bottom panel of Fig. 12 provides a complementary perspective by measuring similarity between corresponding layers, revealing a notable contrast with the global comparison in the top panel of Fig. 12. While overall similarity is moderate when aggregating across layers, the layer-wise analysis shows consistently high similarity (≈0.61–0.95) when comparing aligned depths. This indicates that representations learned by different SSL methods are more comparable at corresponding layers than across arbitrary layers. The discrepancy between the two figures suggests that part of the observed differences across methods arises from cross-layer misalignment rather than uniformly distinct representations. Notably, even methods that appear less similar in the global comparison, such as DINO and MoCoV2, exhibit substantial alignment when evaluated layer-wise (e.g., DINO ≈0.68–0.79, MoCoV2 ≈0.61–0.72), further supporting the view that representation differences are not uniform across depth.

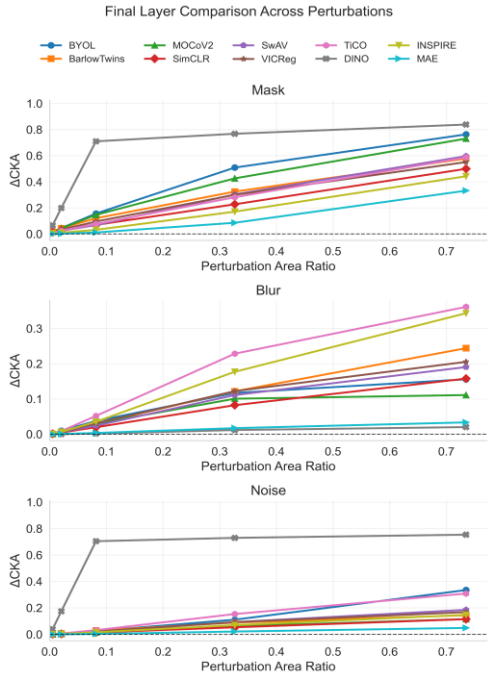


Fig. 13. Final layer representation sensitivity across spatial scales for different SSL models.

Fig. 13 presents the final-layer representation sensitivity across spatial scales for different SSL models under three types of perturbations: masking, blur, and additive noise. For each perturbation, the x-axis denotes the normalized perturbation area ratio, while the y-axis shows the corresponding change in representation measured by ΔCKA. Each curve corresponds to a different SSL method, allowing direct comparison of how representation similarity degrades as the spatial extent of the perturbation increases. The figure summarizes results over 5,000 ImageNet-1K images, providing a stable estimate of model behavior across scales.

Across all perturbation types, a consistent pattern is observed representation change increases with perturbation area, indicating that larger spatial disruptions are associated with greater deviation in learned features. However, the rate and shape of this increase differ substantially across models and perturbation types. Under masking and noise, most models exhibit a largely monotonic increase in ΔCKA, suggesting a progressive accumulation of disruption as more of the image is affected. For example, under masking at an area ratio of 0.7, BYOL reaches approximately ΔCKA ≈ 0.76, while MAE remains substantially lower at around ΔCKA ≈ 0.33, highlighting a clear difference in robustness. In contrast, blur produces a more nuanced response, with several models showing stronger sensitivity at intermediate scales; for instance, TiCo reaches ΔCKA ≈ 0.23 at area ratio 0.3 and further increases to ≈ 0.36 at 0.7, reflecting the selective degradation of mid-frequency structures.

Importantly, models differ in their sensitivity profiles. MAE consistently shows the lowest ΔCKA across all scales and perturbations, indicating strong robustness to both localized and distributed corruption. In contrast, models such as BYOL, TiCo, and MoCoV2 exhibit higher sensitivity, particularly as perturbation extent increases. INSPIRE presents an interesting intermediate behavior: under masking, it shows relatively lower sensitivity at small scales (ΔCKA ≈ 0.04 at 0.1) but increases steadily to ≈ 0.44 at 0.7, consistent with a gradual accumulation of sensitivity. Under blur, INSPIRE is among the most sensitive models at larger scales (reaching ΔCKA ≈ 0.34 at 0.7), suggesting that it is particularly affected by the loss of spatial frequency information. Similarly, under noise, INSPIRE exhibits moderate sensitivity (≈ 0.15 at 0.7), positioning it between highly robust models like MAE and more sensitive ones like BYOL and TiCo.

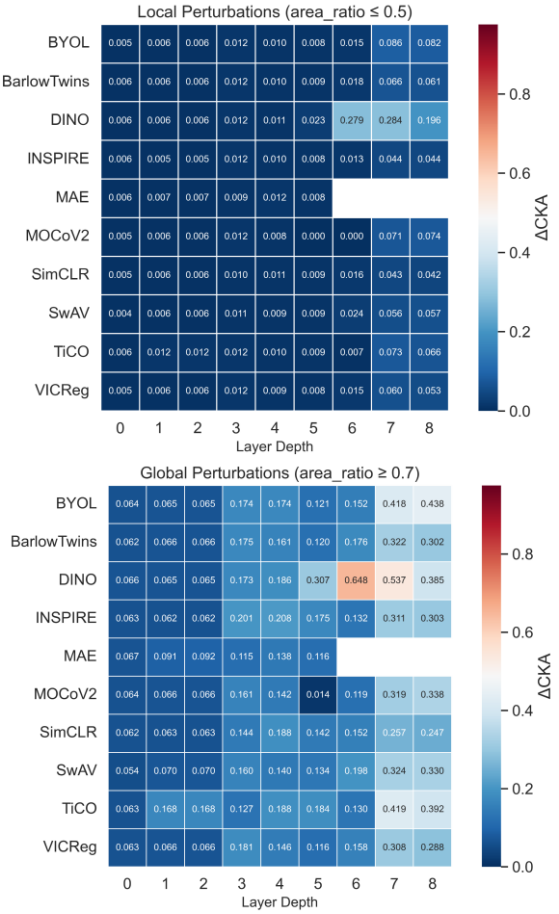


Fig. 14. Layer-wise representation sensitivity (ΔCKA) under local (top) and global (bottom) perturbations.

Fig. 14 presents the layer-wise representation sensitivity (ΔCKA) across models under local and global perturbations. The top panel summarizes local perturbations (area ratio ≤ 0.5), while the bottom panel shows global perturbations (area ratio ≥ 0.7). Each row corresponds to an SSL model, and each column represents a layer indexed by depth. The values indicate the average ΔCKA aggregated over perturbation scales, providing a compact view of how sensitivity evolves across the network hierarchy.

Under local perturbations, all models exhibit very low ΔCKA in early layers (typically ≈ 0.005–0.012), indicating strong stability to small-scale changes at shallow depths. As depth increases, sensitivity gradually rises but remains relatively modest for most models. For example, BYOL increases from ≈ 0.005 in early layers to ≈ 0.086–0.082 in deeper layers, while INSPIRE shows a similar but more controlled increase from ≈ 0.006 to ≈ 0.044. Notably, MAE remains highly stable across all layers, with ΔCKA consistently below ≈ 0.012 in early layers and only reaching ≈ 0.008 before missing deeper entries, reinforcing its robustness to localized perturbations. In contrast, DINO exhibits a sharp increase in deeper layers (≈ 0.279–0.284), suggesting heightened sensitivity to local disruptions at high-level representations.

Under global perturbations, the sensitivity increases significantly across all models, particularly in deeper layers. Early layers remain relatively stable (≈ 0.06–0.09 across models), but deeper layers show substantial divergence. For instance, BYOL increases to ≈ 0.418–0.438 in the deepest layers, while TiCo reaches ≈ 0.419, indicating strong sensitivity to large-scale changes. INSPIRE again exhibits an intermediate behavior: it shows higher sensitivity than MAE but remains more controlled than highly sensitive models, increasing from ≈ 0.063 in early layers to ≈ 0.311–0.303 in deeper layers. This consistent, gradual increase suggests that INSPIRE balances robustness and sensitivity, responding progressively to larger-scale disruptions rather than exhibiting abrupt changes. MAE, in contrast, maintains relatively low sensitivity even under global perturbations (≈ 0.115–0.138 in mid layers), suggesting strong invariance across scales.

The observed sensitivity profiles suggest that SSL representations are neither strictly local nor fully global in their behavior. Representations dominated by localized visual information would be expected to exhibit strong sensitivity to small perturbations, whereas representations dominated by globally aggregated image information would be expected to remain largely unaffected by localized modifications. The measured ΔCKA trends do not fully conform to either extreme. Across multiple methods, representation changes are already observable at small perturbation scales and continue to increase as perturbation extent grows. For instance, DINO exhibits substantial sensitivity even when only a small fraction of the image is perturbed, while methods such as BYOL and SimCLR show progressively larger representation changes as perturbation extent increases (Fig. 13). These observations suggest that SSL representations cannot be adequately described by either strictly local or strictly global forms of spatial encoding. Instead, the results are more consistent with a spectrum of spatial sensitivity behaviors, with different SSL objectives occupying different positions along this continuum.

The experiment also reveals meaningful differences between SSL objectives. DINO exhibits comparatively strong responses even under relatively localized perturbations ($\alpha \leq 0.1$), particularly in deeper layers, suggesting greater sensitivity to spatially structured image content (Fig. 14). In contrast, MAE remains comparatively stable across perturbation scales and network depth (Fig. 13 and Fig. 14), even under large-scale masking. INSPIRE occupies an intermediate position between these extremes: it remains relatively stable under localized

perturbations while progressively increasing in sensitivity at larger scales. Differences across perturbation types further suggest that representation sensitivity depends not only on perturbation extent, but also on the structural nature of the perturbations. In particular, blur consistently produces substantially smaller representation changes than masking, with blur sensitivity peaking near ΔCKA ≈ 0.35 while masking approaches ≈ 0.8 for several models (Fig. 13). This observation suggests that representations remain comparatively stable when coarse spatial structure is preserved, even though fine-grained image details are degraded.

The layer-wise analysis further suggests that spatial integration evolves progressively across network depth (Fig. 14). Early layers remain comparatively stable under both local and global perturbations, whereas deeper layers exhibit substantially stronger sensitivity, particularly for DINO, BYOL, and TiCo. The increasing perturbation sensitivity observed in deeper layers is consistent with progressively broader spatial integration across the network hierarchy, where later representations depend on information distributed across larger spatial extents, a behavior that has been widely observed in hierarchical vision models and attributed to increasing effective receptive field size and contextual feature integration across depth [55].

The results suggest that SSL methods do not converge to a single form of spatial invariance. Different SSL objectives exhibit distinct sensitivity profiles across perturbation scales and network depth, while masking, blur, and noise produce markedly different ΔCKA trends (Fig. 13 and Fig. 14). These consistent differences suggest that spatial sensitivity is an objective-dependent property of the learned representation. Rather than sharing a common notion of invariance, SSL methods appear to occupy different positions on a spatial sensitivity–invariance spectrum.

VII. Computational Cost and Runtime Analysis

In this section, we describe the computational setup and report run-time results for both single- and multi-GPU pretraining experiments. Our goal is to provide a clear account of the hardware environment, training infrastructure, and parallelization strategy to ensure reproducibility and to contextualize the reported run-time characteristics.

All experiments were conducted on a high-performance computing environment consisting of seven compute nodes. Each node is equipped with dual Intel Xeon Silver 4314 processors (16 cores per CPU), 503 GB of system memory, and one NVIDIA A100-PCIE-40GB GPU. Single-GPU experiments were run on a single node using one A100 GPU, with CPUs used for data loading and preprocessing. For multi-GPU pretraining, we employed PyTorch DDP [56] in a multi-node configuration with one GPU per node, where each process handled a portion of the mini-batch and gradients were synchronized across nodes at every iteration. This setup enabled efficient scaling while maintaining consistent optimization behavior across experiments.

A. Single-GPU Pretraining Experiments

For the ImageNet-100 experiments with a ResNet-18 backbone, we adopt a single-GPU training configuration to ensure controlled and consistent comparisons across all variants. Each experiment is executed on a node equipped with an NVIDIA A100 GPU with 40 GB of memory, which provides sufficient capacity for the batch size and augmentation pipeline used in this study. Training is performed for 100 epochs with a batch size of 256 per GPU. Under this configuration, all INSPIRE variants and the SimCLR baseline are trained using identical hardware and resource allocation, ensuring that performance differences are not influenced by variations in computational setup. Data loading and preprocessing are handled on the CPU, while all forward and backward computations are performed on the GPU.

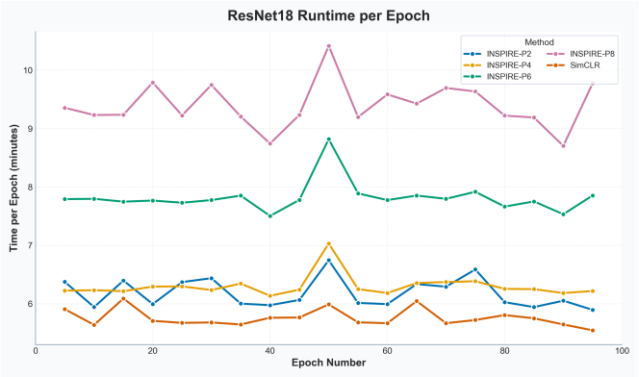


Fig. 15. Training time per epoch for SimCLR and INSPIRE with different patch configurations using a ResNet18 backbone.

Fig. 15 presents the training runtime per epoch for SimCLR and the proposed INSPIRE method under different patch configurations (P = 2, 4, 6, and 8) using a ResNet18 backbone. The training time remains stable across epochs for all configurations, indicating consistent computational behavior throughout the optimization process. SimCLR exhibits the lowest runtime, requiring approximately 5.7 minutes per epoch on average. For INSPIRE, the runtime increases gradually with the number of patches. Specifically, INSPIRE-P2 requires roughly 6.1 minutes per epoch, while INSPIRE-P4 requires about 6.3 minutes. Larger patch configurations introduce additional computational overhead, with INSPIRE-P6 and INSPIRE-P8 requiring approximately 7.8 minutes and 9.3 minutes per epoch, respectively. These results highlight the trade-off between model complexity and computational cost: increasing the number of patches leads to higher training time due to the additional feature processing required by the merge mechanism. Nevertheless, the runtime remains stable across epochs, suggesting that the proposed framework scales predictably with respect to the number of patches.

B. Multi-GPU Parallel Pretraining Experiments

For the ImageNet-1K experiment, we train two variants of INSPIRE with batch sizes 128 and 256 and compare them against SimCLR using a ResNet-50 backbone. All models are trained using PyTorch DDP across 7 GPUs (A100-PCIE-40GB), where each GPU processes its share of the global batch and gradients are synchronized at every step. This setup ensures consistent optimization while maximizing parallel efficiency across nodes.

Fig. 16 presents the training time per epoch (in minutes) on the y-axis versus the epoch number (1–100) on the x-axis. From

the figure, SimCLR consistently runs between approximately 7 and 10 minutes per epoch, with very small fluctuations across training. In comparison, INSPIRE, with batch size 256, shows epoch times that are slightly higher but still relatively close, generally falling in the range of ~11 to 13 minutes as well, with minor variability and a brief warm-up period in the early epochs. This suggests that INSPIRE (256) achieves a comparable level of efficiency to SimCLR in terms of per-epoch runtime. On the other hand, INSPIRE, with batch size 128, operates at a significantly higher cost, with epoch times consistently around ~24 to 26 minutes, and a noticeable spike reaching approximately 27 minutes in one of the early epochs. This clear separation highlights the impact of batch size: reducing the batch size effectively doubles the number of iterations per epoch, leading to a substantial increase in runtime and slightly higher variability.

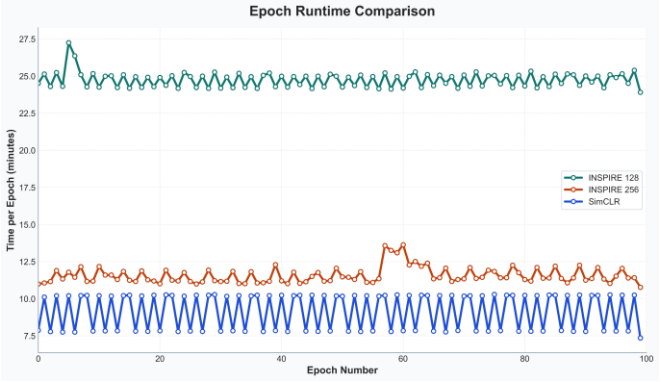


Fig. 16. Training time per epoch for SimCLR and INSPIRE with different batch sizes using a ResNet50 backbone (parallel runtime on 7 GPUs).

These per-epoch trends are consistent with the total training times. SimCLR completes training in 15.27 hours, while INSPIRE Method with batch size 256 takes 19.29 hours, reflecting its slightly higher or comparable per-epoch cost over the full run. In contrast, INSPIRE, with batch size 128, requires 41.26 hours, more than twice as long due to the inefficiency introduced by the smaller batch size.

Scaling from a single-GPU to a multi-node distributed setup plays a critical role in enabling large-scale training on ImageNet-1K. In our experiments, ImageNet-100 training with a ResNet-18 backbone on a single A100 GPU requires approximately 5.7 minutes per epoch for SimCLR, with INSPIRE variants ranging from about 6.1 to 9.8 minutes depending on configuration. In contrast, for the significantly larger ImageNet-1K dataset, we employ PyTorch DDP across 7 A100 GPUs, where each GPU processes a portion of the global batch and gradients are synchronized at every step. Under this setup, SimCLR achieves per-epoch times of approximately 7–10 minutes, while INSPIRE with batch size 256 operates at around 11–13 minutes per epoch. Despite the roughly 10× increase in dataset size, the per-epoch training time increases only modestly, demonstrating the effectiveness of distributed scaling in maintaining practical training times. These results demonstrate that multi-GPU distributed training is essential for scaling contrastive learning to ImageNet-1K, effectively compressing what would otherwise be prohibitively long training times into a manageable range of ~15–19 hours, while maintaining stable and predictable computational behavior across epochs.

VIII. Discussion and Conclusion

This work presents the INSPIRE Framework, a systematic approach for investigating local-global representation learning in self-supervised learning through controlled spatial granularity manipulation during training, downstream evaluation, and representation interpretability using scale-controlled spatial perturbation analysis. While previous research has improved SSL through new objectives [12], [28], architectures [13], [23], and local representation learning strategies [8], [9], it has remained difficult to isolate how training design influences the balance between local and global representations. Our results extend existing studies on representation learning and interpretability by providing a unified experimental framework for systematically analyzing these relationships across multiple evaluation protocols. Although INSPIRE does not achieve the highest performance on every benchmark, it remains consistently competitive across diverse evaluation settings while, more importantly, providing new insights into local-global representation learning, representation sensitivity, and the behavior of learned representations across diverse evaluation protocols.

A central finding of this work is that local-global representation learning is better understood as a continuum than as a binary distinction. The experiments suggest that representations are not inherently local or global; instead, different SSL training strategies occupy different positions along a continuum that reflects how local and global information is balanced within the learned representation. Methods achieving comparable ImageNet performance often exhibit markedly different behavior on fine-grained recognition tasks and under controlled perturbations, supporting that similar global semantic performance can arise from substantially different representational characteristics. This observation provides a more nuanced perspective on SSL representation learning than conventional benchmark comparisons alone.

The controlled spatial granularity experiments further show that the relationship between granularity and representation quality is non-monotonic. Increasing the number of merged regions initially improves representation quality, but excessive fragmentation eventually degrades both global and fine-grained transfer performance. The consistently strong performance of the intermediate P4 configuration suggests that effective representation learning emerges from balancing contextual diversity with semantic coherence rather than maximizing either property independently. This finding indicates that spatial granularity itself represents an important design variable for SSL, independent of network architecture or learning objective.

The combined downstream and representation-level analyses further demonstrate that representation similarity and representation behavior capture complementary aspects of learned representations. Models exhibiting relatively similar representation geometry do not necessarily respond similarly to controlled spatial perturbations, indicating that representation similarity alone provides an incomplete characterization of learned features. Likewise, methods displaying very different perturbation sensitivity may achieve comparable downstream

accuracy. These observations suggest that understanding SSL representations requires considering both the structure of representation space and how representations respond to controlled changes in the input, providing a richer characterization than benchmark accuracy or similarity analysis alone.

The relationship between perturbation sensitivity and downstream transfer also appears more complex than a simple trade-off between invariance and sensitivity. Neither highly sensitive nor highly invariant representations consistently produce the strongest fine-grained recognition performance. Instead, successful transfer appears to depend on maintaining discriminative local information while remaining robust to irrelevant visual variation. This observation reinforces the importance of balanced representations and suggests that representation behavior should be considered alongside conventional evaluation metrics when assessing SSL methods.

Beyond the empirical findings, the proposed INSPIRE Framework has broader implications for SSL research. Rather than introducing only another training method, the framework provides a general methodology for systematically investigating how training strategies influence representation learning. By integrating controlled training, downstream evaluation, and representation interpretability within a unified framework, INSPIRE offers a reusable experimental approach that can support future investigations of emerging SSL methods, alternative architectures, and different application domains. More broadly, the work contributes to the growing shift toward understanding learned representations rather than evaluating them solely through downstream benchmark performance.

Future research can extend the INSPIRE Framework in several directions. The proposed representation interpretability investigation methodology can be complemented with additional representation analysis techniques, including SVCCA, PWCCA, RSA, and probing-based methods, to provide broader perspectives on learned representations. Extending the evaluation to dense prediction tasks such as object detection, semantic segmentation, and instance segmentation would further clarify how local-global representation learning influences spatially intensive vision problems. Future studies should also investigate whether the observed local-global representation continuum and spatial sensitivity characteristics generalize across Vision Transformers, hybrid architectures, multimodal foundation models, and other emerging SSL paradigms. Future work should also investigate the influence of batch size, since the contrastive SSL baselines considered in this work use substantially different effective batch sizes, ranging from 512 to several thousand samples, while INSPIRE employs a fixed effective batch size of 896. We hope that INSPIRE provides a useful foundation for future research on representation learning, interpretability, and the principled design and evaluation of self-supervised learning methods.

ACKNOWLEDGMENT

This article is derived from a research grant funded by the Research, Development, and Innovation Authority (RDIA), Kingdom of Saudi Arabia, with grant number 12615-iu-2023-IU-R-2-1-EI-. The authors gratefully acknowledge the support of the High-Performance Computing Center (Aziz Supercomputer) at King Abdulaziz University (http://hpc.kau.edu.sa) for providing access to computational resources, including A100 GPUs, which were essential for the distributed data parallel (DDP) training of our models.

REFERENCES

[1] M. H. Mozaffari, “Deep Learning for Computer Vision Application,” Electronics 2025, Vol. 14, vol. 14, no. 14, Jul. 2025, doi: 10.3390/ELECTRONICS14142874.

[2] M. Awais et al., “Foundation Models Defining a New Era in Vision: A Survey and Outlook,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 47, no. 4, pp. 2245–2264, 2025, doi: 10.1109/TPAMI.2024.3506283.

[3] R. Alharthi, R. Mehmood, and A. Albeshri, “A Systematic, Scalable, and Interpretable Mapping of Artificial Intelligence Research in Leukemia Using a Hybrid Machine Learning and Qualitative Framework,” Electronics 2026, Vol. 15, vol. 15, no. 5, Mar. 2026, doi: 10.3390/ELECTRONICS15051078.

[4] T. Uelwer et al., “A survey on self-supervised methods for visual representation learning,” Machine Learning 2025 114:4, vol. 114, no. 4, pp. 111-, Mar. 2025, doi: 10.1007/s10994-024-06708-7.

[5] C.-Y. Hsieh, C.-J. Chang, F.-E. Yang, and Y.-C. F. Wang, “Self-Supervised Pyramid Representation Learning for Multi-Label Visual Analysis and Beyond,” Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2696–2705, 2023.

[6] T. Lebailly and T. Tuytelaars, “Global-local self-distillation for visual representation learning,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 1441–1450.

[7] T. Dang, S. Kornblith, H. T. Nguyen, P. Chin, and M. Khademi, “A Study on Self-Supervised Object Detection Pretraining,” Jul. 2022, [Online]. Available: http://arxiv.org/abs/2207.04186

[8] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, “RegionCL: Exploring Contrastive Region Pairs for Self-supervised Representation Learning,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13693 LNCS, pp. 477–494, 2022, doi: 10.1007/978-3-031-19827-4_28.

[9] T. Xiao, C. J. Reed, X. Wang, K. Keutzer, and T. Darrell, “Region similarity representation learning,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10539–10548.

[10] A. Javidani, M. A. Sadeghi, and B. Nadjar Araabi, “Patch-wise self-supervised visual representation learning: a fine-grained approach,” Signal, Image and Video Processing 2025 19:6, vol. 19, no. 6, pp. 458-, Apr. 2025, doi: 10.1007/S11760-025-04020-Y.

[11] H. Xu, X. Zhang, H. Li, L. Xie, H. Xiong, and Q. Tian, “Seed the Views: Hierarchical Semantic Alignment for Contrastive Representation Learning,” Dec. 2020, [Online]. Available: http://arxiv.org/abs/2012.02733

[12] J.-B. Grill et al., “Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning,” Adv. Neural Inf. Process. Syst., vol. 33, pp. 21271–21284, 2020, Accessed: Mar. 11, 2026. [Online]. Available: https://github.com/deepmind/deepmind-research/tree/master/byol

[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” in International conference on machine learning, Nov. 2020, pp. 1597–1607. Accessed: Mar. 11, 2026. [Online]. Available: https://proceedings.mlr.press/v119/chen20j.html

[14] K. Kalasampath, K. N. Spoorthi, S. Sajeev, S. S. Kuppa, K. Ajay, and A. Maruthamuthu, “A Literature Review on Applications of Explainable Artificial Intelligence (XAI),” IEEE Access, vol. 13, pp. 41111–41140, 2025, doi: 10.1109/ACCESS.2025.3546681.

[15] S. Alghamdi, R. Mehmood, F. A. Alqurashi, and A. Alzahrani, “Paving the Roadmap for XAI and IML in Healthcare: Data-Driven Discoveries and the FIXAIH Framework,” IEEE Access, vol. 13, pp. 174393–174427, 2025, doi: 10.1109/ACCESS.2025.3616353.

[16] J. Colin ELLIS Alicante, L. Goetschalckx, N. Oliver, E. Alicante, and S. Thomas Serre, “Capability/= Interpretability: Human Interpretability of Vision Foundation Models,” 2026, doi: https://doi.org/10.48550/arXiv.2605.20337.

[17] T. G. Grigg, D. Busbridge, J. Ramapuram, and R. W. Apple, "Do Self-Supervised and Supervised Methods Learn Similar Visual Representations?," Dec. 2021, Accessed: Mar. 27, 2026. [Online]. Available: https://arxiv.org/pdf/2110.00528

[18] A. Luthra, P. Mishra, and T. Galanti, "On the Alignment Between Supervised and Self-Supervised Contrastive Learning," Oct. 2025, Accessed: Mar. 27, 2026. [Online]. Available: https://arxiv.org/pdf/2510.08852

[19] W. Oleszkiewicz, D. Basaj, T. Trzciński, and B. Zieliński, "Which Visual Features Impact the Performance of Target Task in Self-supervised Learning?," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13350 LNCS, pp. 331–344, 2022, doi: 10.1007/978-3-031-08751-6_24/SAVE-RESEARCH.

[20] W. Oleszkiewicz et al., "Visual Probing: Cognitive Framework for Explaining Self-Supervised Image Representations," IEEE Access, vol. 11, pp. 13028–13043, 2023, doi: 10.1109/ACCESS.2023.3242982.

[21] M. Caron et al., "Emerging Properties in Self-Supervised Vision Transformers," in In Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 9650–9660. [Online]. Available: https://github.com/facebookresearch/dino

[22] H. Choi, S. Jin, and K. Han, "Adversarial Normalization: I Can Visualize Everything (ICE)," in In Proceedings of the IEEE/cvf conference on computer vision and pattern recognition, 2023, pp. 12115–12124.

[23] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 9726–9735, 2020, doi: 10.1109/CVPR42600.2020.00975.

[24] L. Zheng, B. Jing, Z. Li, H. Tong, and J. He, "Heterogeneous Contrastive Learning for Foundation Models and Beyond," Mar. 2024, [Online]. Available: http://arxiv.org/abs/2404.00225

[25] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense Contrastive Learning for Self-Supervised Visual Pre-Training," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition., 2021, pp. 3024–3033. [Online]. Available: https://git.io/

[26] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments," Adv. Neural Inf. Process. Syst., vol. 33, pp. 9912–9924, 2020, Accessed: Mar. 11, 2026. [Online]. Available: https://github.com/facebookresearch/swav

[27] J. Zhu, R. M. Moraes, S. Karakulak, V. Sobol, A. Canziani, and Y. LeCun, "TiCo: Transformation Invariance and Covariance Contrast for Self-Supervised Visual Representation Learning," Jun. 2022, Accessed: Mar. 11, 2026. [Online]. Available: http://arxiv.org/abs/2206.10698

[28] R. Yoshihashi, S. Nishimura, D. Yonebayashi, Y. Otsuka, T. Tanaka, and T. Miyazaki, "Ladder Siamese Network: a Method and Insights for Multi-level Self-Supervised Learning," in IEEE International Conference on Image Processing (ICIP), 2023, pp. 211–215.

[29] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow Twins: Self-Supervised Learning via Redundancy Reduction," in International conference on machine learning, PMLR, Jul. 2021, pp. 12310–12320. Accessed: Mar. 11, 2026. [Online]. Available: https://proceedings.mlr.press/v139/zbontar21a.html

[30] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning," ICLR 2022 - 10th International Conference on Learning Representations, Jan. 2022, Accessed: Mar. 11, 2026. [Online]. Available: http://arxiv.org/abs/2105.04906

[31] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked Autoencoders Are Scalable Vision Learners," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition., 2022, pp. 16000–16009.

[32] S. Yun, H. Lee, J. Kim, and J. Shin, "Patch-Level Representation Learning for Self-Supervised Vision Transformers," in IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 8354–8363.

[33] Q. Guo, Y. Yu, Y. Jiang, J. Wu, Z. Yuan, and P. Luo, "Multi-Level Contrastive Learning for Dense Prediction Task," Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.02010

[34] Q. Su and S. Ji, "Unsqueeze [CLS] Bottleneck to Learn Rich Representations," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 15100 LNCS, pp. 19–37, 2025, doi: 10.1007/978-3-031-72946-1_2/SAVE-RESEARCH.

[35] Z. Zhang and X. Gong, "Positional Label for Self-Supervised Vision Transformer," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 3, pp. 3516–3524, Jun. 2023, doi: 10.1609/AAAI.V37I3.25461.

[36] E. Peruzzo et al., "Spatial entropy as an inductive bias for vision transformers," Machine Learning 2024 113:9, vol. 113, no. 9, pp. 6945–6975, Jul. 2024, doi: 10.1007/S10994-024-06570-7.

[37] J. Zhou et al., "iBOT: Image BERT Pre-Training with Online Tokenizer," ICLR 2022 - 10th International Conference on Learning Representations, Nov. 2021, Accessed: May 13, 2026. [Online]. Available: https://arxiv.org/pdf/2111.07832

[38] Y. Zhan, Y. Zhao, C. Luo, Y. Zhang, and X. Sun, "Attention-Guided Contrastive Masked Image Modeling for Transformer-Based Self-Supervised Learning," Proceedings - International Conference on Image Processing, ICIP, pp. 2490–2494, 2023, doi: 10.1109/ICIP49359.2023.10222606.

[39] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," Proceedings of the 36th International Conference on Machine Learning (ICML), pp. 3519–3529, 2019, Accessed: Mar. 26, 2026. [Online]. Available: http://proceedings.mlr.press/v97/kornblith19a.html

[40] M. Raghu, J. Gilmer, J. Yosinski, and Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep understanding and improvement," Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 6076–6085, 2017, Accessed: Mar. 27, 2026. [Online]. Available: https://yosinski.com/media/papers/Raghu__2017__SVCCA_Singular_Vector_Canonical_Correlation_Analysis.pdf

[41] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?," proceedings.neurips.ccM Raghu, T Unterthiner, S Kornblith, C Zhang, A DosovitskiyAdvances in neural information processing systems, 2021•proceedings.neurips.cc, Accessed: Mar. 27, 2026. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/hash/652cf38361a209088302ba2b8b7f51e0-Abstract.html

[42] A. Li, A. Efros, and D. Pathak, "Understanding Collapse in Non-contrastive Siamese Representation Learning," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13691 LNCS, pp. 490–505, 2022, doi: 10.1007/978-3-031-19821-2_28/SAVE-RESEARCH.

[43] N. Kalibhat, K. Narang, H. Firooz, M. Sanjabi, and S. Feizi, "Measuring self-supervised representation quality for downstream classification using discriminative features," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 13031–13039, 2024, Accessed: Mar. 27, 2026. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/29201

[44] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in Advances in neural information processing systems, 2016. Accessed: Mar. 27, 2026. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/6b180037abbebea991d8b1232f8a8ca9-Abstract.html

[45] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255, 2009, doi: 10.1109/CVPR.2009.5206848.

[46] M. E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," Proceedings - 6th Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2008, pp. 722–729, 2008, doi: 10.1109/ICVGIP.2008.47.

[47] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs," Proc. CVPR workshop on fine-grained visual categorization (FGVC)., vol. 2, 2011, Accessed: Dec. 23, 2025. [Online]. Available: http://vision.stanford.edu/aditya86/StanfordDogs/

[48] M. Cimpoi, S. Maji, I. Kokkinosécole, S. Mohamed, and A. Vedaldi, "Describing Textures in the Wild," InProceedings of the IEEE conference on computer vision and pattern recognition, pp. 3606–3613, 2014, Accessed: Dec. 23, 2025. [Online]. Available: http://www.robots.ox.ac.uk/

[49] W. Catherine, B. Steve, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, 2011.

[50] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D Object Representations for Fine-Grained Categorization," InProceedings of the IEEE international conference on computer vision workshops, pp. 554–561, 2013, doi: 10.1109/ICCVW.2013.77.

[51] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," Computer Vision–ECCV 2014, vol. 8689, no. PART 1, pp. 818–833, 2014, doi: 10.1007/978-3-319-10590-1_53.

[52] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in International conference on learning representations, 2018. Accessed: Apr. 01, 2026. [Online]. Available: https://github.com/rgeirhos/texture-vs-shape

[53] D. Hendrycks and T. Dietterich, "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations," 7th International Conference on Learning Representations, ICLR 2019, Mar. 2019, Accessed: Apr. 01, 2026. [Online]. Available: https://arxiv.org/pdf/1903.12261

[54] "Benchmarks — LightlySSL 1.5.22 documentation." Accessed: Dec. 24, 2025. [Online]. Available: https://docs.lightly.ai/self-supervised-learning/getting_started/benchmarks.html

[55] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., vol. 29, 2016.

[56] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Neural Information Processing Systems, 2019.