Self-Supervised Representation Learning as Mutual Information Maximization

Anonymous authors
Paper under double-blind review

Abstract

Self-supervised representation learning (SSRL) has demonstrated remarkable empirical success, yet its underlying principles remain insufficiently understood. While recent works attempt to unify SSRL methods by examining their information-theoretic objectives or summarizing their heuristics for preventing representation collapse, architectural elements like predictor networks, stop-gradient operations, and statistical regularizers are often viewed as empirically motivated additions. In this paper, we adopt a first-principles approach and investigate whether the learning objective of an SSRL algorithm dictates its possible optimization strategies and model design choices. In particular, by starting from a variational mutual information (MI) lower bound, we derive two training paradigms, namely Self-Distillation MI (SDMI) and Joint MI (JMI), each imposing distinct structural constraints and covering a set of existing SSRL algorithms. SDMI relies on alternating optimization, in which stop-gradient operations serve as a principled mechanism for realizing the alternating updates. In contrast, JMI admits joint optimization through symmetric architectures without such components. Under the proposed formulation, predictor networks in SDMI and statistical regularizers in JMI emerge as tractable surrogates for the MI objective. We show that many existing SSRL methods are specific instances or approximations of these two paradigms. This paper provides a theoretical explanation for the choices of different architectural components of existing SSRL methods, going beyond heuristic conveniences.

1 Introduction

SSRL has demonstrated remarkable success at learning useful features from unlabeled data, achieving competitive performance with supervised approaches across a wide range of tasks (LeCun et al., 2015; Bengio et al., 2013; Balestriero et al., 2023). Conventionally, SSRL algorithms are divided into two categories according to their training objectives, namely contrastive methods and non-contrastive methods. Contrastive methods (Oord et al., 2018; Tian et al., 2020; Chen et al., 2020a; He et al., 2020; Chen et al., 2020b; 2021) train a representation model by aligning the representations of two augmentations of the same input while explicitly pushing apart those of augmentations of different inputs. Non-contrastive methods (Grill et al., 2020; Chen & He, 2021; Caron et al., 2021; Zbontar et al., 2021; Bardes et al., 2022; Sui et al., 2024), by contrast, challenge the necessity of negative samples and propose alternative mechanisms, such as momentum encoders or stop-gradient operations, to prevent representational collapse.

Many recent studies attempt to unify these two categories of SSRL methods under common theoretical frameworks, often through shared information-theoretic principles. Liu et al. (2022) interpret various SSRL methods as low-order approximations of a unified maximum entropy principle. Zbontar et al. (2021) motivate the Barlow Twins objective via Information Bottleneck theory (Tishby et al., 1999; Tishby & Zaslavsky, 2015), which Tsai et al. (2021) later relate to a kernel-based MI measure. Shwartz-Ziv et al. (2023) connect VICReg's penalties to MI bounds, and Jha et al. (2024) propose a unifying framework that explains collapse avoidance by minimizing a global mean while preserving augmentation-level variation. Despite their insights, prior work offers little clarity on whether training strategies like self-distillation or variance–covariance control are heuristic additions or principled consequences of the objective itself, leaving an important gap in our theoretical understanding of these methods.

In this work, we close this gap by returning to first principles, analyzing existing SSRL algorithms through the lens of MI maximization, a perspective implicitly or explicitly invoked by many self-supervised learning methods. Starting from a variational lower bound on MI, particularly the Donsker–Varadhan (DV) bound, we show that this bound naturally leads to two optimization paradigms in the context of SSRL. The first, Self-Distillation MI (SDMI), uses EM-style alternating updates with stop-gradient operations (e.g., SimSiam, BYOL, MoCo). The second, Joint MI (JMI), supports joint optimization via a single gradient step per batch (e.g., SimCLR, Barlow Twins, VICReg). We further argue that this taxonomy is more theoretically principled than the traditional contrastive vs. non-contrastive distinction, as it is derived directly from the DV bound rather than from heuristic descriptions of training behavior. We then distill SDMI and JMI into canonical algorithmic forms, and demonstrate that these forms behave similarly to existing SSRL methods in their respective paradigms and achieve competitive performance on downstream tasks.

In summary, we make the following contributions.

1. We formulate a general MI maximization perspective under the DV bound, showing that existing SSRL methods implicitly follow one of two optimization paradigms, namely Self-Distillation MI (SDMI) or Joint MI (JMI).
2. We show that design elements such as stop-gradients, predictor networks, and statistical regularizers (typically introduced as empirical fixes) arise as principled consequences of optimizing an MI lower bound under one of the two paradigms, providing a formal explanation for these common design choices.
3. We show that many well-known SSRL approaches (e.g., SimCLR, BYOL, SimSiam) can be mapped directly to our two paradigms. This helps unify a broad set of SSRL methods under a shared theoretical lens and offers guidance for future method design.

2 Related Work and Preliminaries

We begin by reviewing recent attempts to unify the growing landscape of SSRL methods under shared theoretical principles. We first summarize key unification efforts based on objective design and collapse-prevention mechanisms, highlighting their contributions and limitations. We then present MI as a foundational concept and starting point for our analysis, revisiting its definition and variational lower bounds, with a focus on the DV bound, whose tightness and decomposition are central to our work.

2.1 Unification approaches in SSRL

A growing body of work (Zbontar et al., 2021; Liu et al., 2022; Tsai et al., 2021; Shwartz-Ziv et al., 2023; Jha et al., 2024; Tan et al., 2024) suggests that an information-theoretic lens can help unify seemingly disparate SSRL methods. Many existing methods, particularly contrastive approaches, can be explicitly framed as maximizing MI between representations of different augmented views (Oord et al., 2018; He et al., 2020; Chen et al., 2020b; 2021; Poole et al., 2019).

Building on this, several works link specific SSRL losses to MI estimation. The Information Bottleneck perspective (Tishby et al., 1999; Tishby & Zaslavsky, 2015) is applied to Barlow Twins (Zbontar et al., 2021), and Tsai et al. (2021) show that the Barlow Twins objective is equivalent to maximizing a Hilbert–Schmidt Independence Criterion (Gretton et al., 2005), a kernelized dependence measure related to MI. Bardes et al. (2022) introduce VICReg's variance and covariance penalties, and Shwartz-Ziv et al. (2023) later provide an information-theoretic analysis linking these penalties to MI bounds.

Another prominent unification direction is offered by Liu et al. (2022), who propose a Maximum Entropy Coding (MEC) framework that treats representation learning as an entropy maximization problem, showing that many existing SSRL methods can be interpreted as low-order Taylor approximations of a single entropy-based objective. Complementing this view, Jha et al. (2024) analyze the collapse avoidance mechanisms that ensure stability in SSRL, arguing that, despite architectural and algorithmic differences, most methods implicitly minimize the global average of learned representations while preserving sample-level variability.

While these approaches provide valuable unifying perspectives on SSRL objectives or collapse-prevention mechanisms, they do not address whether the commonly used optimization strategies and architectural components, such as stop-gradient operations, predictor networks, or statistical regularizers, are necessary consequences of the learning objective itself or simply heuristic choices.

2.2 Mutual information and its variational bounds in SSRL

In SSRL, MI is often defined between representations Z_A and Z_B of two augmented views X_A and X_B of an input X in the form $I(Z_A; Z_B) = D_{\mathrm{KL}}\left[p(z_A, z_B) \,\|\, p(z_A)p(z_B)\right]$. Maximizing MI with respect to the encoding function $Z_A = f_\theta(X_A)$ defines a valid pretext task for learning representations that can transfer to various downstream applications. However, direct optimization of MI is intractable since the underlying data distribution $P(X)$ is unknown, motivating the use of variational bounds in practice. Common variational bounds include InfoNCE (Oord et al., 2018; Poole et al., 2019), Barber–Agakov (Barber & Agakov, 2003), TUBA (Poole et al., 2019), NWJ (Nguyen et al., 2010), JSD (Hjelm et al., 2019) and DV (Belghazi et al., 2018). Each of these alternatives introduces different trade-offs between tightness, stability, and optimization feasibility.

We use the Donsker–Varadhan (DV) bound to guide our analysis in this paper, as it offers (1) a direct connection to MI via KL divergence, (2) a natural variational decomposition that facilitates block-coordinate ascent, and (3) provable tightness relative to f-divergence-based alternatives for any fixed function class (Belghazi et al., 2018).

Donsker–Varadhan bound Over a sufficiently rich class of functions $\mathcal{F}$, the DV bound decomposes MI as

$$I(Z_A; Z_B) \geq I_{\mathrm{DV}}(Z_A; Z_B) = \sup_{T \in \mathcal{F}} \left\{ \underbrace{\mathbb{E}_{p(z_A, z_B)}[T(z_A, z_B)]}_{\text{Joint term}} - \underbrace{\log \mathbb{E}_{p(z_A)p(z_B)}\left[e^{T(z_A, z_B)}\right]}_{\text{Marginal term}} \right\}, \tag{1}$$

where $\mathcal{F} \subseteq \{ f : \mathcal{Z}_A \times \mathcal{Z}_B \to \mathbb{R} \}$, and $T \in \mathcal{F}$ is a scoring function that assigns high values to joint pairs $(z_A, z_B) \sim p(z_A, z_B)$ and low values to marginal pairs $(z_A, z_B) \sim p(z_A)p(z_B)$.

3 A Unified View of SSRL as MI Maximization

In this section, we first revisit the DV lower bound on MI from an optimization perspective. This gives rise to two natural optimization paradigms in SSRL, namely Self-Distillation MI (SDMI) and Joint MI (JMI). We then analytically show how a wide range of SSRL methods can be categorized under these paradigms.

3.1 Block-coordinate ascent via DV bound

Let representations Z_A and Z_B come from two different encoding functions f_θ and g_ξ with a fixed scoring function T drawn from the function class $\mathcal{F}$. We note that the DV bound in Eq. (1) provides a useful formulation for optimization, since exactly maximizing the bound with respect to θ while holding ξ fixed (and vice versa) guarantees a non-decreasing improvement of the objective. As a result, alternating updates over the encoders for Z_A and Z_B constitute valid block-coordinate ascent steps. Specifically, we formalize the improvement in the following proposition.

Proposition Let the DV-bound objective be given by

$$\mathcal{L}(\theta, \xi) = J(\theta; \xi) - M(\theta; \xi), \tag{2}$$

where $\mathcal{L}(\theta, \xi)$ is the DV bound, $J(\theta; \xi)$ is the joint term, and $M(\theta; \xi)$ is the marginal term from Eq. (1).

Assume that (1) for fixed ξ, $J(\cdot; \xi)$ is concave in θ; (2) the marginal term $M(\cdot; \xi)$ is smooth and satisfies $\|\nabla_\theta M(\theta; \xi)\| \leq \varepsilon$; and (3) the same conditions hold symmetrically for updates over ξ. Then alternating gradient steps over θ and ξ yield approximate monotonic improvement in $\mathcal{L}(\theta, \xi)$:

$$\mathcal{L}(\theta^{(k+1)}, \xi^{(k)}) \geq \mathcal{L}(\theta^{(k)}, \xi^{(k)}) - \mathcal{O}(\varepsilon), \quad \mathcal{L}(\theta^{(k+1)}, \xi^{(k+1)}) \geq \mathcal{L}(\theta^{(k+1)}, \xi^{(k)}) - \mathcal{O}(\varepsilon).$$

See Section A.1 for our proof. In particular, if $\varepsilon \to 0$ (e.g., slowly changing marginal distributions), the objective becomes asymptotically non-decreasing over iterations.

When sharing parameters $\theta = \xi$, the maximization objective in Eq. (2) can be jointly optimized via standard gradient ascent with the guarantee of monotonic improvement, provided that the full objective $\mathcal{L}(\theta)$ is concave. In the DV bound (Eq. (1)), the joint term is affine and the marginal term is convex with respect to the scoring function T (as a log-partition function), providing a principled motivation for joint gradient ascent when $\theta = \xi$.

The same block-coordinate argument extends to other variational MI bounds with the joint-minus-marginal structure, such as InfoNCE and JSD; see Section A.2.

As such, there are two valid optimization paradigms to maximize MI: alternating updates across encoder branches or joint updates over shared parameters. We name the two paradigms Self-Distillation MI (SDMI) and Joint MI (JMI), respectively.

3.2 Self-Distillation Mutual Information (SDMI)

SSRL methods in the SDMI paradigm rely on an EM-style alternating update schedule between two encoder branches and on a mechanism for maximizing MI between augmented views. The alternating updates are enabled through a stop-gradient operator, which breaks the gradient flow from one branch to the other, making it possible to treat one encoder as fixed while updating the other, mimicking a block-coordinate ascent on the DV bound. Typically, these methods use an online encoder that receives direct gradient updates and a target (or momentum) encoder that is updated via an exponential moving average (EMA) of the online encoder's parameters. While some existing SDMI methods such as SimSiam and BYOL do not explicitly optimize a variational MI bound, we show that their alternating update structure, enabled by stop-gradients and architectural asymmetry, can be derived as a principled optimization strategy for DV-bound maximization. This provides a theoretical justification for design choices that were previously considered heuristic.

Block-coordinate interpretation of SDMI To formalize SDMI as an EM-style block-coordinate ascent procedure, we consider batches of two augmented views $X_1 = \{x_1^i\}_{i=1}^N$ and $X_2 = \{x_2^i\}_{i=1}^N$, where each x_1^i, x_2^i is sampled from a stochastic augmentation $\mathcal{A}(x)$ applied to an input $x \sim P(x)$. We denote the batch size by N and use two encoders f_θ and g_ξ.

E-Step: At iteration k, we define the MI between the representations produced by the encoders f_θ and g_ξ as

$$I^{(k)} = I\big(f_{\theta^{(k)}}(X_1),\ g_{\xi^{(k)}}(X_2)\big). \tag{3}$$

We update the f_θ encoder by maximizing the objective under a stop-gradient (SG) on the g_ξ encoder:

$$\theta^{(k+1)} = \arg\max_{\theta}\ I\big(f_\theta(X_1),\ \mathrm{SG}(g_{\xi^{(k)}}(X_2))\big), \tag{4}$$

which guarantees $I\big(f_{\theta^{(k+1)}}(X_1),\ g_{\xi^{(k)}}(X_2)\big) \geq I\big(f_{\theta^{(k)}}(X_1),\ g_{\xi^{(k)}}(X_2)\big)$.

M-Step: Using the updated f_θ encoder, we update the g_ξ encoder with a stop-gradient on f_θ,

$$\xi^{(k+1)} = \arg\max_{\xi}\ I\big(\mathrm{SG}(f_{\theta^{(k+1)}}(X_1)),\ g_\xi(X_2)\big), \tag{5}$$

ensuring $I\big(f_{\theta^{(k+1)}}(X_1),\ g_{\xi^{(k+1)}}(X_2)\big) \geq I\big(f_{\theta^{(k+1)}}(X_1),\ g_{\xi^{(k)}}(X_2)\big)$.

Monotonic Improvement: Together, these steps guarantee overall monotonic improvement:

$$I\big(f_{\theta^{(k+1)}}(X_1),\ g_{\xi^{(k+1)}}(X_2)\big) \geq I\big(f_{\theta^{(k+1)}}(X_1),\ g_{\xi^{(k)}}(X_2)\big) \geq I\big(f_{\theta^{(k)}}(X_1),\ g_{\xi^{(k)}}(X_2)\big). \tag{6}$$

3.2.1 Examples of SDMI methods

SimSiam and BYOL Non-contrastive methods, such as BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021), fall under the SDMI paradigm. These methods train an online encoder f_θ, together with a

lightweight predictor h_ϕ, to align transformed representations with those of a target encoder g_ξ. From the SDMI viewpoint, both methods approximate a two-step EM-style optimization in a relaxed, implicit form, as described below.

1. **E-step:** Both methods update the online encoder to maximize MI by minimizing the following negative cosine similarity loss

$$\min_{\theta,\phi} \; -\mathbb{E}_{p(x_1,x_2)}\big[T_{\cos}\big(h_\phi(f_\theta(x_1)),\, g_\xi(x_2)\big)\big], \tag{7}$$

   where $T_{\cos}$ denotes a cosine similarity scoring function. This loss can be viewed as an instantiation of the DV bound with cosine similarity, which we refer to as $I_{\text{cos-DV}}$. However, these methods omit the explicit marginal term present in the full bound (see Eq. (14)), relying instead on their predictor dynamics to discourage collapse.

2. **M-step (Implicit):** Immediately after the E-step, SimSiam resets the target encoder with the new online weights and freezes it for the next E-step:

$$g_{\text{new}} = \text{SG}(f_\theta). \tag{8}$$

   BYOL, on the other hand, uses an EMA of θ:

$$\xi \leftarrow \tau\xi + (1-\tau)\theta. \tag{9}$$

   While these methods differ from SDMI's explicit coordinate ascent step on the g_ξ encoder, they preserve the underlying principle of alternating optimization, though in an implicit form.

Our interpretation aligns with the hypothesis of Chen & He (2021) that SimSiam's stop-gradient induces EM-like alternating updates between online and frozen branches. While they suggest that the predictor approximates an expectation over augmentations, Zhang et al. (2022) refute this, showing instead that it induces de-centering and de-correlation gradients that stabilize training and promote feature diversity. Within our SDMI framework, we reinterpret these effects as implicitly approximating the marginal term of the DV bound. Section 4.3 provides further analysis and empirical evidence supporting this interpretation.

MoCo MoCo (He et al., 2020; Chen et al., 2020b; 2021), a contrastive learning method, also fits naturally within the SDMI paradigm. It performs EM-style alternating updates between an online encoder and a momentum encoder, while directly optimizing the InfoNCE lower bound on MI. Its momentum encoder plays a similar functional role and is updated via EMA, like the target encoder in BYOL. Early versions (MoCo-v1 (He et al., 2020), v2 (Chen et al., 2020b)) already achieve strong performance without predictor networks, and although MoCo-v3 (Chen et al., 2021) introduces a predictor, it yields only marginal performance gains ($\sim 1\%$), underscoring that with direct MI maximization, predictors are auxiliary.

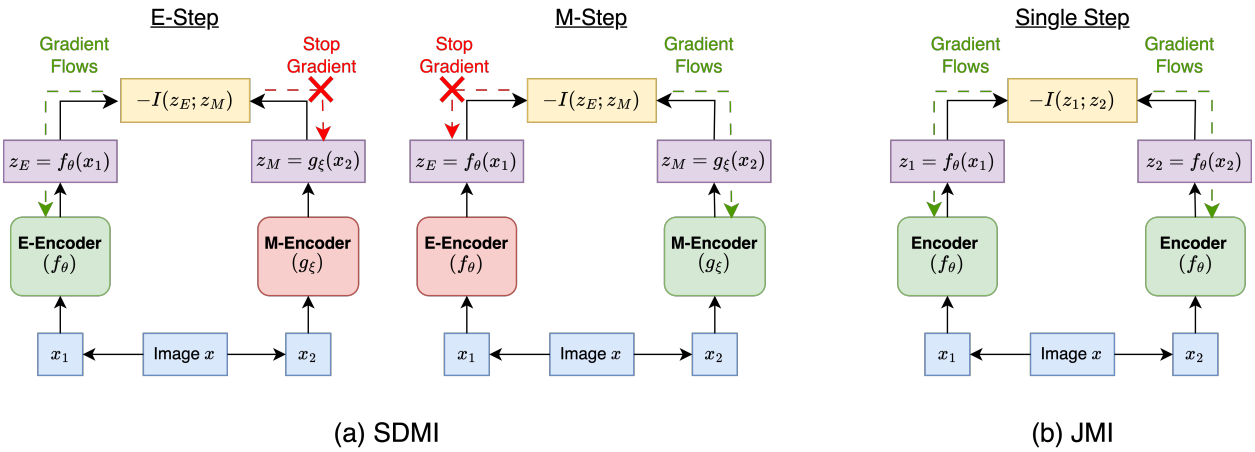
This illustrates how the SDMI framework unifies both traditional contrastive and non-contrastive methods under a shared lens of MI maximization with alternating encoder updates.

3.3 Joint Mutual Information (JMI)

Unlike SDMI, JMI methods use a single encoder f_θ to produce representations for both augmented views, enabling joint gradient updates to maximize MI. Joint optimization is achieved either by directly optimizing an explicit MI objective or by incorporating surrogate regularization terms that penalize statistical properties, such as variance, covariance, or feature redundancy, to approximate the marginal log-partition term in Eq. (1). A general JMI objective written as a loss function takes the form

$$\mathcal{L}_{\text{JMI}} = -\mathbb{E}_{p(x_1,x_2)}\left[T(f_\theta(x_1), f_\theta(x_2))\right] + \lambda \cdot \mathcal{R}(f_\theta(x_1), f_\theta(x_2)). \tag{10}$$

Examples of JMI methods include contrastive learning methods such as SimCLR, which directly optimizes InfoNCE to maximize MI between views. More recent non-contrastive methods, such as Barlow Twins (Zbontar et al., 2021) and VICReg (Bardes et al., 2022), optimize an alignment term between augmented views together with a regularizer that approximates the marginal term from Eq. (1).



(a) SDMI (b) JMI

Figure 1: Canonical forms of our proposed paradigms: (a) SDMI alternates updates between two encoders using stop-gradients, while (b) JMI jointly updates both views with shared gradients.

3.4 From DV to Barlow Twins: A surrogate derivation

To show how non-contrastive methods can be seen as using feature-level regularization as in Eq. (10), we demonstrate how the Barlow Twins loss could be derived from Eq. (1) using several straightforward approximations and assumptions, providing a direct connection between the Barlow Twins loss and mutual information maximization between views. To begin, we replace the DV bound's marginal term with its second-order Taylor approximation[1]:

$$\mathcal{L}_{\text{Taylor-DV}} = -\underbrace{\mathbb{E}_{p(z_A,z_B)}[T(z_A,z_B)]}_{\text{Joint term}} + \underbrace{\mathbb{E}_{p(z_A)p(z_B)}[T(z_A,z_B)]}_{\text{Marginal mean term}} + \underbrace{\text{Var}_{p(z_A)p(z_B)}[T(z_A,z_B)]}_{\text{Marginal variance term}}. \tag{11}$$

Barlow Twins corresponds to the particular choice of the dot product scoring function,

$$T(z^A, z^B) = \sum_{i=1}^{d} z_i^A z_i^B, \tag{12}$$

which is an approximation to the optimal T in Eq. (1). Since batch normalization is normally applied, we also assume $\mathbb{E}[z_i^A] \approx \mathbb{E}[z_i^B] \approx 0$, which effectively removes the marginal mean term from Eq. (11), leaving the alignment and variance terms as the primary components.

Barlow Twins is usually expressed with the empirical cross-correlation matrix,

$$C_{ij} = \frac{1}{N} \sum_{n=1}^{N} z_{n,i}^A \cdot z_{n,j}^B,$$

where alignment is encouraged via the diagonal C_{ii}, and redundancy is penalized via the off-diagonals C_{ij}, $i \neq j$. To simplify the variance term in Eq. (11), we expand it using the dot product in Eq. (12), and further assume jointly Gaussian representations with decorrelation within each view, which then implies (by Isserlis' theorem (Munthe-Kaas et al., 2025))

$$\text{Var}\left[\sum_i z_i^A z_i^B\right] \approx \sum_{i \neq j} C_{ij}^2.$$

[1]The $\frac{1}{2}$ coefficient that appears in front of the variance term in the standard second-order Taylor expansion of the log-MGF is omitted here for simplicity, as it can be absorbed into a tuning hyperparameter.

Putting together these components yields a moment-based surrogate to the DV objective

$$\mathcal{L}_{\text{Taylor-DV}} \approx -\sum_i C_{ii} + \sum_{i \neq j} C_{ij}^2, \tag{13}$$

which closely matches the Barlow Twins loss. We provide the full derivation in Section A.3.

In summary, SDMI and JMI represent two principled optimization paradigms for maximizing MI. Our findings reveal that many architectural components in modern SSRL methods, previously introduced as heuristic choices, can instead be interpreted as structured consequences of optimizing MI. We illustrate the distinction between SDMI and JMI in Figure 1, and provide a summary of representative SSRL methods in Table 7, together with their classification under the SDMI/JMI taxonomy, including whether they employ explicit MI objectives or surrogate regularizers.

4 Experiments

This section empirically validates the theoretical structure of SDMI and JMI by instantiating their canonical forms and analyzing their behavior alongside representative SSRL methods. The purpose of this study is not to suggest that the canonical forms of SDMI and JMI are state-of-the-art SSRL methods. Instead, we use them as a simplified setting to understand the dynamics of MI training and representation quality, and to examine how the optimization principles derived from MI manifest in practice. We compare the canonical forms to more specialized and performant variants from the literature to shed light on the role of MI maximization in SSRL.

4.1 Canonical SDMI and JMI prototypes

To empirically validate the theoretical framework developed in Section 3, we instantiate minimal, controlled implementations of the SDMI and JMI paradigms. These *canonical forms* exclude auxiliary components such as momentum updates, predictor networks, or regularizers, and serve to isolate the optimization structure derived from the DV bound. As illustrated in Figure 1, SDMI alternates updates between two encoders using stop-gradients, while JMI applies symmetric joint updates to both augmented views using a shared encoder. Both prototypes optimize the same MI objective defined below, enabling a direct comparison of their dynamics.

Objective: Cosine-based DV bound While the DV bound is theoretically maximally tight when $\mathcal{F}$ is a sufficiently broad class of functions, in practice, unrestricted neural critics T often lead to high variance and unstable training behavior (Oord et al., 2018; Poole et al., 2019; Song & Ermon, 2020). To ensure reliable estimation while preserving the validity of the DV bound, we restrict the critic function T to cosine similarity, $T(z_A, z_B) = \frac{z_A \cdot z_B}{\|z_A\|_2 \|z_B\|_2}$, providing a stable, bounded, and scale-invariant surrogate. This choice is further motivated by its widespread use in SSRL objectives (Chen et al., 2020a; He et al., 2020; Chen et al., 2020b; 2021; Grill et al., 2020; Chen & He, 2021), where it serves as a standard metric for comparing representations across augmented views. By restricting T to be the cosine similarity, we effectively optimize only over the representations of Z_A and Z_B:

$$\begin{aligned} I(Z_A; Z_B) \geq I_{\text{DV}}(Z_A; Z_B) \geq I_{\text{cos-DV}}(Z_A; Z_B) \\ = \mathbb{E}_{p(z_A, z_B)}\left[T_{\cos}(z_A, z_B)\right] - \log \mathbb{E}_{p(z_A)p(z_B)}\left[e^{T_{\cos}(z_A, z_B)}\right]. \end{aligned} \tag{14}$$

Although using $I_{\text{cos-DV}}$ sacrifices some tightness, it provides a more stable estimator while remaining a lower bound of the MI objective.

Practical approximation To compute the marginal term in Eq. (14) efficiently, we approximate the expectation using off-diagonal cross-pairs from a batch of size N:

$$\log \mathbb{E}_{p(z_A)p(z_B)}\left[e^{T_{\cos}(z_A, z_B)}\right] \approx \log\left(\frac{1}{N(N-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{N} e^{T_{\cos}(z_A^{(i)}, z_B^{(j)})}\right).$$

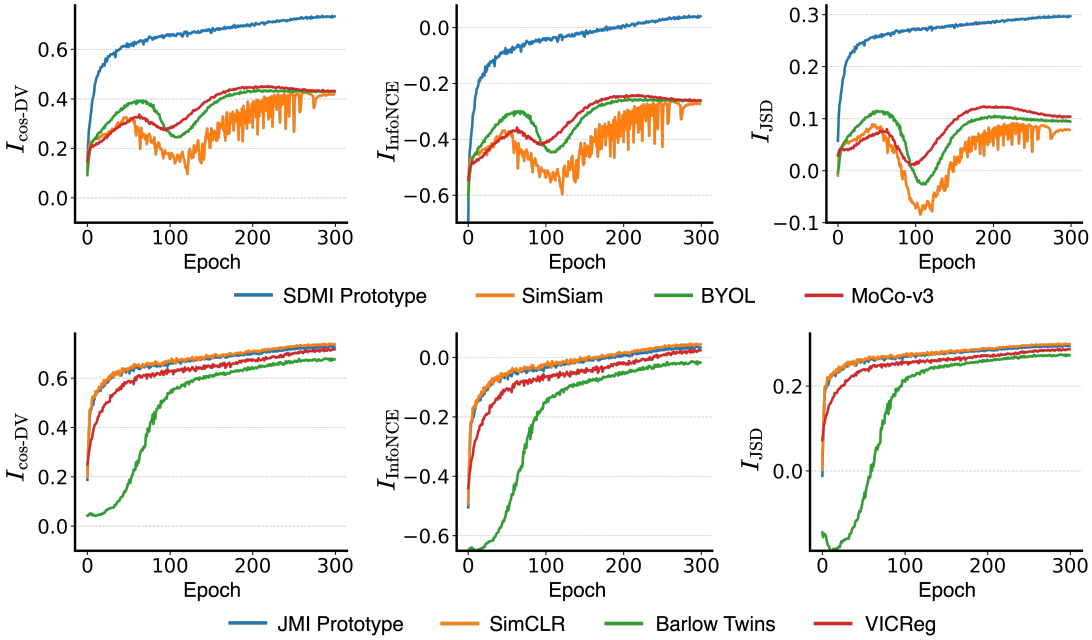


Figure 2: Estimated MI over CIFAR10 training for SDMI-based (top row) and JMI-based (bottom row) methods, using three estimators (cos-DV, InfoNCE, and JSD; left to right). Both paradigms exhibit consistent MI growth: SDMI curves feature early fluctuations before trending upward, while JMI estimates rise more uniformly and to much higher levels.

Hence, our batchwise training objective takes the form:

$$\mathcal{L}_{\text{cos-DV}} = -\left[\frac{1}{N}\sum_{i=1}^{N} T_{\cos}(z_A^{(i)}, z_B^{(i)}) - \log\left(\frac{1}{N(N-1)}\sum_{\substack{i,j=1\\ i\neq j}}^{N} e^{T_{\cos}(z_A^{(i)}, z_B^{(j)})}\right)\right]. \tag{15}$$

Full pseudocode for the SDMI and JMI training procedures is provided in Algorithms 1 and 2 of Section E.

4.2 Experimental setup

Datasets We use standard SSRL benchmarks, including CIFAR10/100 (Krizhevsky & Hinton, 2009), Tiny-ImageNet, and ImageNet100 (Deng et al., 2009). Additionally, for controlled experiments and visualization, we generate a toy dataset from a mixture of five isotropic Gaussian distributions centered at evenly spaced points on the unit circle. Each cluster center is defined by $\mu_k = \left(\cos\frac{2\pi k}{5}, \sin\frac{2\pi k}{5}\right), \quad k = 1, \ldots, 5$, with samples drawn as $x \sim \mathcal{N}(\mu_k, \sigma^2 I)$, where $\sigma = 0.05$ and I is the 2×2 identity matrix. Two augmented views are generated by perturbing x with independent Gaussian noise: $x_1 = x + \epsilon_1, \quad x_2 = x + \epsilon_2, \quad \epsilon_1, \epsilon_2 \sim \mathcal{N}(0, \tau^2 I)$, where $\tau = 0.1$. We generate $N = 2500$ samples, with 500 samples per cluster.

Implementation details We implement the canonical SDMI prototype (Figure 1(a)) with two independently initialized encoders trained via alternating E- and M-step updates. The JMI prototype (Figure 1(b)) uses a single shared encoder updated jointly with symmetric gradients. All baseline methods use their standard architectures and objectives. Our canonical SDMI and JMI prototypes use ResNet-18 (He et al., 2016) encoders for CIFAR10/100 and TinyImageNet, and ResNet-50 encoders for ImageNet100 (Deng et al., 2009). For the Gaussian dataset, each method (including all baselines) uses a dedicated two-layer MLP encoder mapping $\mathbb{R}^2 \to \mathbb{R}^3$: a linear layer ($2 \to 64$) with bias, batch normalization, and ReLU activation, followed by a second linear layer ($64 \to 3$) with batch normalization, with the output normalized to unit ℓ_2-norm.

Table 1: Linear probing accuracy (%) on four datasets. Mean ± std over 3 runs.

Model	CIFAR10	CIFAR100	TinyImageNet	ImageNet100
SDMI prototype (f_θ)	88.61 ±0.13	57.37 ±0.38	33.30 ±0.58	70.73 ±0.57
SDMI prototype (g_ξ)	88.59 ±0.35	57.85 ±0.32	32.94 ±0.71	70.83 ±0.16
SimSiam	89.72 ±0.18	60.45 ±0.60	19.19 ±0.69	78.23 ±0.58
BYOL	91.28 ±0.16	63.11 ±0.21	32.77 ±0.10	81.09 ±0.61
MoCo-v3	91.10 ±0.16	58.90 ±0.32	32.18 ±0.55	76.86 ±0.74
JMI Prototype	88.01 ±0.48	57.22 ±0.56	32.23 ±0.52	73.41 ±0.36
SimCLR	87.24 ±0.37	55.32 ±0.46	33.79 ±0.31	75.31 ±0.76
Barlow Twins	85.56 ±0.71	51.91 ±0.49	30.26 ±0.12	78.96 ±0.30
VICReg	85.49 ±1.03	54.00 ±0.34	32.03 ±0.32	78.86 ±0.23

Mutual information estimation To assess MI dynamics during training, we compute three variational bounds: the cos-DV bound ($I_{\text{cos-DV}}$) from Eq. (14), the InfoNCE bound (I_{InfoNCE}) (Oord et al., 2018; Poole et al., 2019), and the JSD bound (I_{JSD}) (Hjelm et al., 2019).

For JMI-based methods (JMI prototype, SimCLR, Barlow Twins, and VICReg), both augmented views are passed through the same encoder f_θ, and MI is computed between the representations:

$$I^{(t)} = I\big(f_\theta^{(t)}(x_1),\, f_\theta^{(t)}(x_2)\big).$$

For SDMI-based methods (SDMI prototype, SimSiam, BYOL, MoCo-v3), MI is measured between two asymmetric encoder branches. In the SDMI prototype, these are independently updated f_θ and g_ξ encoders trained via alternating updates:

$$I^{(t)} = I\big(f_\theta^{(t)}(x_1),\, g_\xi^{(t)}(x_2)\big).$$

In BYOL and MoCo-v3, g_ξ is a momentum encoder updated via EMA. In SimSiam, which lacks a persistent target encoder, we instead treat the previous epoch's encoder state as the M-branch:

$$I_{\text{SimSiam}}^{(t)} = I\big(f_\theta^{(t)}(x_1),\, f_\theta^{(t-1)}(x_2)\big),$$

with $I^{(0)} = -\infty$ by convention.

4.3 Results

Monotonic MI increase Figure 2 shows estimated MI over training for all methods across both paradigms on the CIFAR10 dataset. Since the SDMI prototype explicitly optimizes the cos-DV bound in Eq. (14), while MoCo-v3 optimizes InfoNCE, the JSD bound serves as an independent estimator not optimized by any method. Compared to the other SDMI methods, the SDMI prototype (top row) exhibits a near-perfect monotonic increase in MI throughout training. This is expected, as it explicitly optimizes the cos-DV bound (Eq. (14)) using true EM-style alternating updates between two independently parameterized encoders. In contrast, methods like SimSiam, BYOL, and MoCo only approximate this behavior through their architectural heuristics, which lead to a noisy MI estimate and generally lower final MI levels. Nevertheless, all methods still exhibit an overall upward MI trend, confirming that they retain the underlying MI-maximization structure. Meanwhile, all JMI-based methods display smooth and consistently increasing MI curves, reflecting their symmetric joint-update optimization. We provide additional results on the Gaussian data in Section 4.4, confirming this trend in ideal conditions.

Cluster center trajectories in embedding space To visualize how well the representation space separates underlying structure, we use the Gaussian dataset and track the movement of all five cluster centers during training (Figure 3). We quantify separation via the nearest-neighbor (NN) angle gap, the mean angular distance to the closest other center. The JMI prototype achieves the largest separation, reaching an average NN gap of $\approx 88°$, with the closest JMI-based competitor at $\approx 82°$. The SDMI prototype also shows

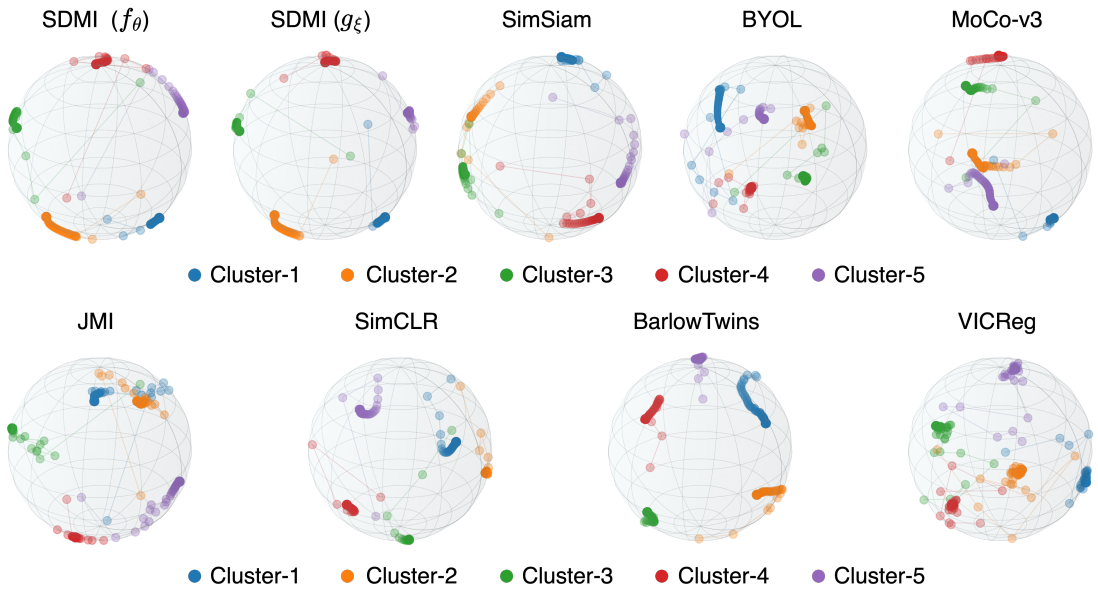


Figure 3: Embedding trajectories of the five Gaussian cluster centers. Opacity increases over training, showing how different methods progressively separate the clusters in embedding space.

strong separation, with an average NN angle gap of $\approx 74°$, compared to $\approx 50°$ for the strongest SDMI-based competitor. Detailed metrics and comparative analysis are presented in Section B.1.

Linear probing To assess the quality of learned representations for downstream tasks, we perform linear probing on real-world datasets. We train encoders, freeze them, and then train a linear classifier head using cross-entropy (Tian et al., 2020). As shown in Table 1, our prototype methods are competitive with established SSRL methods across both SDMI and JMI paradigms. No single method outperforms all others consistently. See Section C for implementation details.

It is worth noting that our canonical SDMI and JMI models are intentionally minimal, showing that theory-driven models can provide strong baselines without the need for empirically driven architectural tweaks like predictor heads, EMA, or regularization. Existing SSRL methods can be viewed as augmenting such minimal forms with task-specific architectural improvements. Our work focuses on explanation rather than optimization.

Predictors as DV-bound marginal surrogates

Predictor networks and stop-gradients are widely recognized as essential components in SDMI-based SSRL methods (Chen & He, 2021; Balestriero et al., 2023; Jha et al., 2024), with several works analyzing their role in preventing collapse (Zhang et al., 2022; Tian et al., 2021; Srinath Halvagal et al., 2023; Shi et al., 2020; Wang et al., 2021). In Section 3.2, we show that stop-gradient enables the EM-style alternating optimization. Zhang et al. (2022) prove that the predictor prevents collapse by decomposing its gradient into center and residual components, showing it induces de-centering and dimensional de-correlation, mechanisms equivalent to those from negative samples in contrastive learning. We complement this understanding with a controlled ablation study on CIFAR10 using linear probing (Figure 4(a)), systematically adding and removing components from the SDMI prototype. Removing the marginal term from Eq. (1) results in representational collapse, despite EM-style updates, confirming the necessity of a complete MI objective. More generally, in the absence of the marginal regularization term, encoders converge to trivial solutions.

Remarkably, adding a trainable predictor to the f_θ encoder during the E-step and to the g_ξ encoder during the M-step, while still omitting the marginal term, entirely prevents collapse. This behavior suggests that predictors function as implicit surrogates for the log-partition (marginal) term, validating our analysis in Section 3.2.1 that predictors recover the missing normalization component in the MI maximization objective.

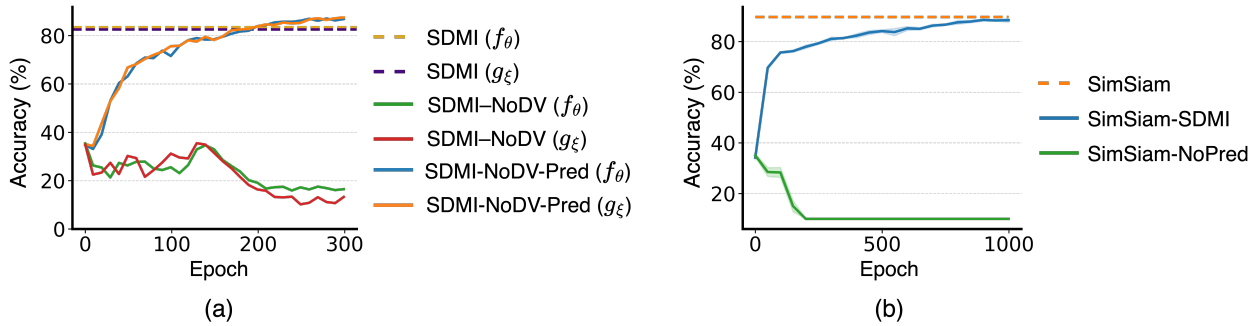


Figure 4: **(a) Linear probing accuracy of SDMI prototype under ablations on CIFAR10.** Removing the marginal term from the DV bound leads to representational collapse in both the f_θ and g_ξ encoders, denoted as SDMI-NoDV (f_θ) and SDMI-NoDV (g_ξ), respectively, despite the use of alternating (EM-style) optimization, confirming the necessity of the marginal regularization term. Remarkably, adding trainable predictors during the E-step and M-step, SDMI-NoDV-Pred (f_θ) and SDMI-NoDV-Pred (g_ξ), while still omitting the marginal term, entirely prevents collapse and recovers strong performance. Dashed lines represent baseline accuracy of the pure SDMI prototype with all components intact. **(b) SimSiam variants under different loss functions and predictor configurations on CIFAR10.** Removing the predictor leads to collapse, confirming that SimSiam's cosine loss alone lacks marginal regularization. Replacing the loss with the explicit cos-DV objective (SimSiam-SDMI) restores performance without requiring a predictor.

Using SimSiam as an example, we now examine what happens when a non-contrastive method within the SDMI paradigm is given an explicit MI objective. First, we remove the predictor from SimSiam while retaining its alternating update scheme. This variant collapses, since minimizing cosine similarity is equivalent to optimizing only the joint term of the DV bound (Eq. (1)), with no marginal correction to prevent trivial solutions. Next, we replace SimSiam's heuristic loss with the cos-DV bound-based loss from Eq. (15), while still using a single-encoder alternating schedule but without the predictor. We call this variant SimSiam-SDMI. Remarkably, SimSiam-SDMI not only avoids collapse, but also recovers nearly the original linear probe accuracy (baseline: $89.68 \pm 0.35\%$, SimSiam-SDMI: $88.43 \pm 0.78\%$). Figure 4(b) summarizes these trends across all three variants.

Discussion Interestingly, while the SDMI prototype achieves the highest MI under all three bounds, and the most separated clusters of representations, this does not translate directly into higher downstream performance. This suggests that maximizing MI, though necessary to prevent representational collapse, is not by itself sufficient for optimal SSRL performance. MI should thus be viewed as a foundation rather than the ultimate objective of SSRL. Crucially, our results show that the optimization paradigm, SDMI or JMI, and the strategies and components it uses, determine how the MI objective is approximated and, in turn, the usefulness of the learned features for downstream tasks.

In summary, *how* MI is optimized matters as much as *how much* MI is achieved. By formalizing the SDMI and JMI paradigms and identifying their essential components, our taxonomy provides a roadmap for future research. We recommend that future efforts prioritize designing better strategies and architectural components tailored to each optimization paradigm, thereby better bridging the gap between MI maximization and downstream task performance.

4.4 MI estimation

To better understand the dynamics of MI maximization without confounding factors from complex image data, we repeat our MI-tracking experiment on the controlled synthetic dataset and toy MLP encoder described in Section 4.2. In this setting, we can isolate the effect of optimization since the ground-truth data distribution is simple and noise is well-characterized. We compute the same three variational bounds in a deterministic setting (i.e., a single batch update): the cos-DV bound ($I_{\text{cos-DV}}$) from Eq. (14), the InfoNCE bound (I_{InfoNCE}) (Oord et al., 2018; Poole et al., 2019), and the JSD bound (I_{JSD}) (Hjelm et al., 2019). At

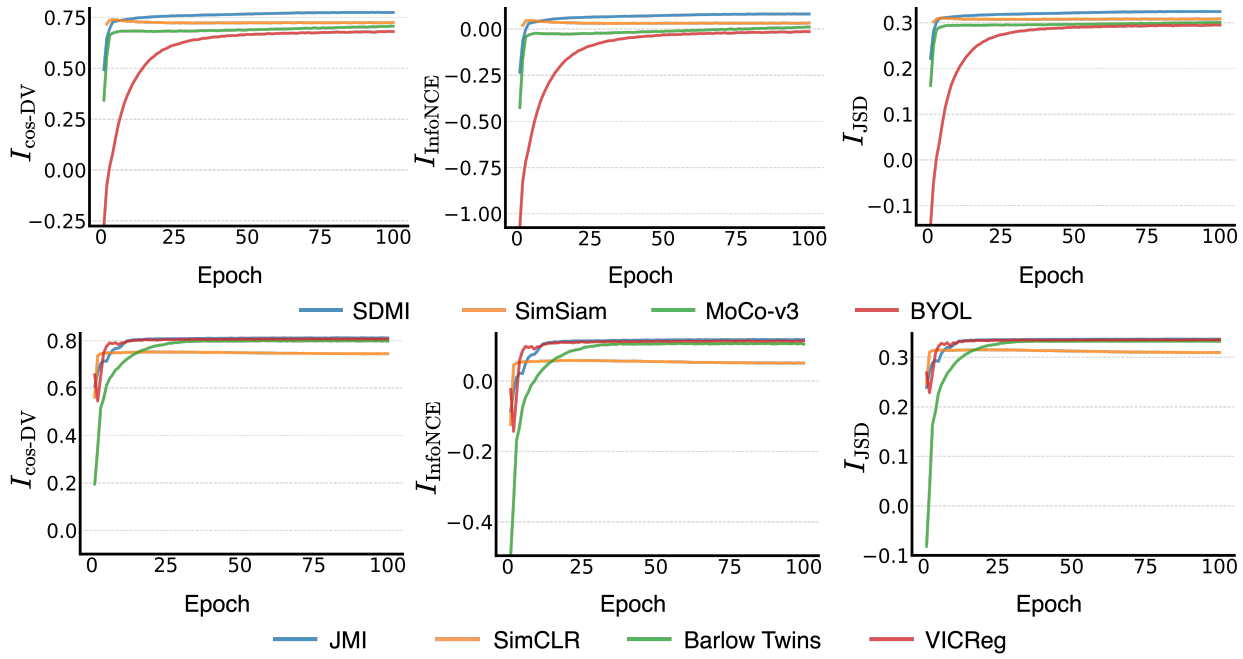


Figure 5: Estimated MI over training using cos-DV, InfoNCE, and JSD bounds for both SDMI methods (top) and JMI methods (bottom). All three estimators show approximately monotonic growth for all methods under both paradigms.

each epoch, we compute all three MI estimates on frozen encoder outputs from a validation set consisting of 2,500 data points. As shown in Figure 5, all three bounds across all methods show similar, near-monotonic MI increase during training.

5 Conclusion

In this work, we revisited SSRL from first principles, grounding our analysis in a variational MI optimization lens. By deriving two distinct training paradigms, SDMI and JMI, we showed that many design choices in SSRL architectures are not merely empirical conveniences but theoretically motivated necessities. By unifying a broad class of existing SSRL methods under the theoretical lens, our analysis offers an alternative understanding of the mechanisms that drive successful representation learning and guides the principled design of future SSRL algorithms.

Limitations

While our framework offers a principled view of SSRL via MI maximization, our experiments are limited to image datasets. Extending the analysis to other data modalities such as text, audio, or multimodal settings would strengthen the generalizability of our theoretical insights and is a promising direction for future work.

Broader Impact Statement

By clarifying the principles behind self-supervised learning, this work may support more robust and interpretable model design. Though theoretical, our findings could influence the development of trustworthy AI systems in socially impactful domains.

References

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson,

Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning. *arXiv:2304.12210*, 2023.

David Barber and Felix Agakov. Information Maximization in Noisy Channels : A Variational Approach. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. In *International Conference on Learning Representations*, 2022.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 531–540, 2018.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

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

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 1597–1607, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

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

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9640–9649, 2021.

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

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 3015–3024, 2021.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring Statistical Dependence with Hilbert-Schmidt Norms. In *Algorithmic Learning Theory*, pp. 63–77. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-31696-1.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

Abhishek Jha, Matthew B Blaschko, Yuki M Asano, and Tinne Tuytelaars. The common stability mechanism behind most self-supervised learning approaches. *arXiv:2402.14957*, 2024.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

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

Hans Z Munthe-Kaas, Olivier Verdier, and Gilles Vilmart. A short proof of Isserlis' theorem. *arXiv:2503.01588*, 2025.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11): 5847–5861, 2010.

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

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 5171–5180, 2019.

Haizhou Shi, Dongliang Luo, Siliang Tang, Jian Wang, and Yueting Zhuang. Run away from your teacher: Understanding BYOL by a novel self-supervised approach. *arXiv:2011.10944*, 2020.

Ravid Shwartz-Ziv, Randall Balestriero, Kenji Kawaguchi, Tim G. J. Rudner, and Yann LeCun. An information theory perspective on variance-invariance-covariance regularization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 33965–33998, 2023.

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2020.

Manu Srinath Halvagal, Axel Laborieux, and Friedemann Zenke. Implicit variance regularization in non-contrastive SSL. In *Advances in Neural Information Processing Systems*, volume 36, pp. 63409–63436, 2023.

Yi Sui, Tongzi Wu, Jesse C. Cresswell, Ga Wu, George Stein, Xiao Shi Huang, Xiaochen Zhang, and Maksims Volkovs. Self-supervised representation learning from random data projectors. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhiquan Tan, Jingqin Yang, Weiran Huang, Yang Yuan, and Yifan Zhang. Information flow in self-supervised learning. In *Forty-first International Conference on Machine Learning*, 2024.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. In *International Conference on Learning Representations*, 2022.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 776–794. Springer, 2020.

Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 10268–10278, 2021.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop*, pp. 1–5, 2015. doi: 10.1109/ITW.2015.7133169.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377, 1999.

Yao-Hung Hubert Tsai, Shaojie Bai, Louis-Philippe Morency, and Ruslan Salakhutdinov. A note on connecting Barlow twins with negative-sample-free contrastive learning. *arXiv:2104.13712*, 2021.

Xiang Wang, Xinlei Chen, Simon S Du, and Yuandong Tian. Towards demystifying representation learning with non-contrastive self-supervision. *arXiv:2110.04947*, 2021.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 12310–12320, 2021.

Chaoning Zhang, Kang Zhang, Chenshuang Zhang, Trung X. Pham, Chang D. Yoo, and In So Kweon. How Does SimSiam Avoid Collapse Without Negative Samples? A Unified Understanding with Self-supervised Contrastive Learning. In *International Conference on Learning Representations*, 2022.

A Further analysis

A.1 Block-coordinate ascent in MI bounds

We provide the formal proof for the proposition stated in Section 3.1, establishing the theoretical foundation for monotonic MI increase under alternating optimization in the SDMI paradigm.

Proof. Fix $\xi^{(k)}$. Since $J(\cdot;\xi^{(k)})$ is concave, a gradient ascent step on θ with sufficiently small step size guarantees

$$J(\theta^{(k+1)};\xi^{(k)}) \geq J(\theta^{(k)};\xi^{(k)}). \tag{16}$$

By smoothness of M and the bound $\|\nabla_\theta M\| \leq \varepsilon$, we have

$$\left|M(\theta^{(k+1)};\xi^{(k)}) - M(\theta^{(k)};\xi^{(k)})\right| = \mathcal{O}(\varepsilon), \tag{17}$$

which yields

$$\mathcal{L}(\theta^{(k+1)},\xi^{(k)}) \geq \mathcal{L}(\theta^{(k)},\xi^{(k)}) - \mathcal{O}(\varepsilon). \tag{18}$$

An identical argument applies to the ξ-update. Chaining the two completes the proof. □

A.2 Analyzing other variational bounds

Extending the analysis from Section 3.1, we examine other commonly used variational MI bounds in SSRL, namely the InfoNCE and JSD bounds mentioned in Section 2.2, demonstrating that our framework generalizes beyond the DV bound.

A.2.1 InfoNCE

Recall that the InfoNCE loss between two representations Z_A and Z_B takes the form

$$\begin{aligned}\mathcal{L}_{\text{InfoNCE}} &= -\mathbb{E}_{p(z_A,z_B)}\left[\log\left(\frac{e^{T(z_A,z_B)}}{\sum_{z'_B} e^{T(z_A,z'_B)}}\right)\right] \\ &= -\mathbb{E}_{p(z_A,z_B)}\left[T(z_A,z_B) - \log\left(\sum_{z'_B} e^{T(z_A,z'_B)}\right)\right],\end{aligned} \tag{19}$$

where T is a similarity function and the inner sum runs over the negative samples z'_B in a batch.

This loss can be interpreted as a lower bound on MI between Z_A and Z_B (Poole et al., 2019), such that

$$I_{\text{InfoNCE}}(Z_A;Z_B) = \mathbb{E}_{p(z_A,z_B)}[T(z_A,z_B)] - \mathbb{E}_{p(z_A)}\left[\log \mathbb{E}_{p(z_B)}\left[e^{T(z_A,z_B)}\right]\right] + \log N, \tag{20}$$

where N is the number of negative samples.

Both the DV and the InfoNCE bounds follow the general structure of a joint term minus a marginal term. The only structural difference is in how the logarithm is aggregated: the DV bound aggregates globally before applying the logarithm,

$$\log \mathbb{E}_{p(z_A)p(z_B)}\left[e^{T(z_A,z_B)}\right] \quad \text{(global aggregation)},$$

whereas InfoNCE applies the logarithm per sample,

$$\mathbb{E}_{p(z_A)}\left[\log \mathbb{E}_{p(z_B)}\left[e^{T(z_A,z_B)}\right]\right] \quad \text{(local aggregation)}.$$

Although this difference affects the aggregation structure, both objectives satisfy the conditions of the proposition in Section 3.1 and support monotonic improvement under alternating optimization.

A.2.2 JSD

Similarly, the JSD bound can be expressed as

$$I_{\mathrm{JSD}}(Z_A; Z_B) = \mathbb{E}_{p(z_A,z_B)}\big[-\log\big(1+e^{-T(z_A,z_B)}\big)\big] - \mathbb{E}_{p(z_A)p(z_B)}\big[\log\big(1+e^{T(z_A,z_B)}\big)\big].$$

This form corresponds to a binary classification objective, distinguishing samples from the joint distribution versus the product of marginals. As with DV and InfoNCE, it has a *joint term minus marginal term* structure, but instead of a log-sum-exp aggregation, it applies the softplus nonlinearity independently to each sample.

All three objectives (DV, InfoNCE, JSD) satisfy the conditions of the proposition in Section 3.1 and admit monotonic improvement under alternating optimization.

A.3 Moment-based surrogates for the DV marginal term

We present the complete mathematical derivation referenced in Section 3.4 showing that the regularizers in Barlow Twins correspond to a second-order Taylor expansion (cumulant expansion) of the DV bound's marginal term.

DV bound

To ground our approximation, we recall the DV bound (Eq. (1)):

$$I(Z_A; Z_B) \geq I_{\mathrm{DV}}(Z_A; Z_B) = \sup_{T\in\mathcal{F}} \left\{ \underbrace{\mathbb{E}_{p(z_A,z_B)}[T(z_A,z_B)]}_{\text{Joint term}} - \underbrace{\log \mathbb{E}_{p(z_A)p(z_B)}\Big[e^{T(z_A,z_B)}\Big]}_{\text{Marginal term}} \right\},$$

where $T \in \mathcal{F}$ is a critic function, chosen from a sufficiently expressive function class $\mathcal{F}$.

CGF and Taylor expansion

Let T be any bounded critic with

$$T(x,y) \in [a,b] \quad \text{for all } x, y,$$

and define its CGF as

$$K(s) = \log \mathbb{E}[e^{sT}]. \tag{21}$$

Because T is bounded, K is infinitely differentiable on $[0,1]$. Its n-th derivative at zero yields the n-th cumulant,

$$\kappa_n = K^{(n)}(0),$$

and in particular,

$$K'(0) = \mathbb{E}[T], \quad K''(0) = \mathrm{Var}(T).$$

By Taylor's theorem about $s = 0$, for $s \in [0,1]$,

$$K(s) = s\,K'(0) + \tfrac{1}{2}s^2\,K''(0) + R_2(s), \tag{22}$$

where $R_2(s) = \frac{1}{6}s^3 K^{(3)}(s^*)$ for some $s^* \in (0,s)$.

Since $T \in [a,b]$, all derivatives of $K(s)$ are bounded on $[0,1]$. In particular, evaluating Eq. (22) at $s = 1$ yields

$$\big|R_2(1)\big| \leq \frac{1}{6}\max_{s\in[0,1]}\big|K^{(3)}(s)\big| = \mathcal{O}(1).$$

The residual is therefore bounded, and the second-order approximation

$$\log \mathbb{E}[e^T] = K(1) \approx \mathbb{E}[T] + \tfrac{1}{2}\mathrm{Var}(T) \tag{23}$$

holds with controlled error.

Surrogate loss via mean–variance

Substituting Eq. (23) into the DV bound (Eq. (1)) gives the moment-based surrogate

$$\begin{aligned} I_{\mathrm{DV}}(Z_A; Z_B) \;&\approx\; I_{\text{Taylor-DV}}(Z_A; Z_B) \\ &= \mathbb{E}_{p(z_A,z_B)}[T(z_A, z_B)] - \left\{\mathbb{E}_{p(z_A)p(z_B)}[T(z_A, z_B)] + \tfrac{1}{2}\,\mathrm{Var}_{p(z_A)p(z_B)}\big[T(z_A, z_B)\big]\right\}. \end{aligned} \tag{24}$$

We may construct a tractable loss as in Eq. (11)

$$\mathcal{L}_{\text{Taylor-DV}} = -\underbrace{\mathbb{E}_{p(z_A,z_B)}[T(z_A, z_B)]}_{\text{Joint term}} + \underbrace{\mathbb{E}_{p(z_A)p(z_B)}[T(z_A, z_B)]}_{\text{Marginal mean term}} + \underbrace{\mathrm{Var}_{p(z_A)p(z_B)}[T(z_A, z_B)]}_{\text{Marginal variance term}},$$

where the $\frac{1}{2}$ coefficient on the variance term is omitted for simplicity, as it can be absorbed into a tuning hyperparameter.

Barlow Twins as a mean–variance surrogate

We use z_i^A and z_i^B for the i-th coordinate of views A and B, respectively. Starting from Eq. (12),

$$X_i = z_i^A\, z_i^B, \quad T_{\cos}(z^A, z^B) = \sum_{i=1}^{d} X_i = z^A \cdot z^B.$$

By the variance-of-a-sum identity,

$$\mathrm{Var}\big[T(z^A, z^B)\big] = \mathrm{Var}\left(\sum_{i=1}^{d} X_i\right) = \sum_{i,j=1}^{d} \mathrm{Cov}(X_i,\, X_j) = \sum_{i,j=1}^{d} \mathrm{Cov}(z_i^A z_i^B,\; z_j^A z_j^B). \tag{25}$$

Barlow Twins reduces the surrogate in Eq. (11) to an alignment term and a tractable approximation of the marginal variance by applying batch normalization, ensuring

$$\mathbb{E}_{p(z_A)}[z_i^A] = \mathbb{E}_{p(z_B)}[z_i^B] \approx 0 \quad \Rightarrow \quad \mathbb{E}_{p(z_A)p(z_B)}[T(z_A, z_B)] \approx 0.$$

The regularization terms in Barlow Twins are constructed using batch-level statistics, specifically the *cross-correlation matrix* between features across the two views,

$$C_{ij} = \frac{1}{N}\sum_{n=1}^{N} z_{n,i}^A \cdot z_{n,j}^B,$$

where $z_{n,i}^A$ and $z_{n,j}^B$ denote the i-th and j-th features of the n-th sample from each view in a batch of size N. The diagonal elements C_{ii} appear in the alignment term of the loss, encouraging each feature to match across views, while the off-diagonal elements C_{ij} for $i \neq j$ are penalized to reduce redundancy.

To connect this to the variance term in Eq. (11), we analyze the variance of Eq. (12) under independent sampling,

$$\mathrm{Var}_{p(z_A)p(z_B)}\left[\sum_{i=1}^{d} z_i^A z_i^B\right] = \sum_{i,j} \mathrm{Cov}(z_i^A z_i^B,\; z_j^A z_j^B). \tag{26}$$

Expanding this covariance using its definition gives a fourth-order moment,

$$\mathrm{Cov}(X_i, X_j) = \mathbb{E}[z_i^A z_i^B\; z_j^A z_j^B] - \mathbb{E}[z_i^A z_i^B] \cdot \mathbb{E}[z_j^A z_j^B]. \tag{27}$$

Assuming that the representations are approximately jointly Gaussian and decorrelated within each view (i.e., $\mathbb{E}[z_i^A z_j^A] \approx 0$, $\mathbb{E}[z_i^B z_j^B] \approx 0$ for $i \neq j$), we apply Isserlis' theorem (Munthe-Kaas et al., 2025) to approximate the fourth-order covariance terms,

$$\mathrm{Cov}(z_i^A z_i^B,\; z_j^A z_j^B) \approx \mathbb{E}[z_i^A z_j^B] \cdot \mathbb{E}[z_j^A z_i^B] = C_{ij} C_{ji} \approx C_{ij}^2, \quad \text{for } i \neq j. \tag{28}$$

The variance thus approximates the sum of off-diagonal squared correlations,

$$\mathrm{Var}\left[\sum_i z_i^A z_i^B\right] \approx \sum_{i\neq j} C_{ij}^2.$$

Putting everything together, the Taylor–DV surrogate yields Eq. (13),

$$\mathcal{L}_{\text{Taylor-DV}} \approx -\sum_{i=1}^{d} C_{ii} + \sum_{i\neq j} C_{ij}^2,$$

which matches the structure of the empirical Barlow Twins loss: an alignment term encouraging the diagonal of the cross-correlation matrix to approach 1, and a decorrelation term penalizing off-diagonal elements.

A.4 Input informativeness

While the SDMI and JMI frameworks increase $I(z_E; z_M)$, their effectiveness depends on how this relates to the input x. We formalize this intuition with the following conjecture.

Proposition (view-MI is bounded by input informativeness). Assume that, conditional on the input x, the two augmented views x_1, x_2 are drawn independently from the augmentation distribution $A(x)$, so that $z^{(1)} \perp\!\!\!\perp z^{(2)} \mid x$. Then

$$I(z^{(1)}; z^{(2)}) \;\leq\; \min\{\, I(x; z^{(1)}),\; I(x; z^{(2)}) \,\}. \tag{29}$$

Proof. Under the stated assumption, $z^{(1)} \leftrightarrow x \leftrightarrow z^{(2)}$ forms a Markov chain. The Data Processing Inequality then gives

$$I(z^{(1)}; z^{(2)}) \;\leq\; I(x; z^{(1)}),$$

and by symmetry $I(z^{(1)}; z^{(2)}) \leq I(x; z^{(2)})$. Combining the two inequalities yields

$$I(z^{(1)}; z^{(2)}) \;\leq\; \min\{\, I(x; z^{(1)}),\; I(x; z^{(2)}) \,\}. \quad \square$$

B Additional experimental results

Supplementary experimental results and analyses support the findings presented in Section 4, including controlled experiments on synthetic data and additional ablation studies.

B.1 Controlled experiment

B.1.1 SDMI deterministic full-batch updates

Nearest-neighbor angle statistics. To complement Figure 3, we report rotation-invariant NN angle statistics for the final cluster embeddings in Table 2. In our setup, on a 3D unit sphere, the ideal separation for five clusters corresponds to a NN gap of $\approx 90^\circ$ (Thomson optimum).

Cluster dynamics. In Figure 6, we plot the embeddings of each centroid under both f_θ and g_ξ encoders at various training iterations (epochs $1, 5, 20, 70, 100$). As training progresses, the two views of each cluster increasingly align with one another, while the embeddings across different clusters become progressively more separated, indicating that the independently updated encoders learn both consistent and discriminative representations.

Table 2: NN angle gaps at convergence.

Model	Mean NN Gap (°)	Min NN Gap (°)	Max NN Gap (°)	SD (°)
SDMI prototype (Encoder f_θ)	**74.47**	**72.78**	**77.60**	1.76
SDMI prototype (Encoder g_ξ)	73.77	72.16	75.06	**1.32**
SimSiam	40.62	28.04	71.86	16.19
MoCo	49.64	47.90	54.07	2.29
BYOL	37.35	31.26	45.34	5.42
JMI prototype	**88.26**	**85.49**	**94.03**	**3.36**
SimCLR	65.80	61.95	70.60	3.38
Barlow Twins	81.62	64.46	102.53	15.55
VICReg	82.15	69.28	100.95	14.54

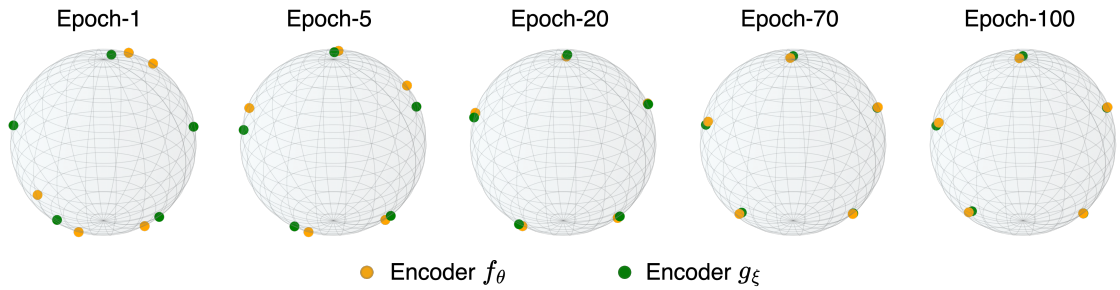


Figure 6: Cluster centers on the unit sphere, showing how SDMI prototype encoders progressively separate the five Gaussian clusters across training.

B.1.2 SDMI stochastic mini-batch updates

To examine whether the SDMI prototype continues to maximize MI under stochastic optimization, we rerun the toy Gaussian mixture experiment using a training batch size of 500 and standard SGD (learning rate 0.5, cosine annealing schedule, 100 epochs). Figure 7 shows the estimated $I_{\text{cos-DV}}$, I_{InfoNCE}, and I_{JSD} curves for SDMI and the three baseline methods. All approaches continue to show approximately monotonic MI growth despite the use of mini-batches.

C Implementation details

C.1 Compute resources

All experiments were conducted on two servers equipped with an NVIDIA RTX 5090 GPU and an NVIDIA RTX 4080 GPU. The complete set of experiments, including hyperparameter sweeps and additional experiments not included in the paper, took approximately 6 weeks of wall-clock time. For the SDMI prototype, GPU memory (VRAM) usage was around 15 GB, while all other methods required approximately 7 GB.

C.2 Real data benchmark

Hyperparameter sweep 1. We conduct a grid search for each model using a ResNet-18 encoder backbone on the CIFAR10/100 datasets. The hyperparameters explored in the initial sweep are summarized in Table 3. Models requiring a predictor network (e.g., SimSiam, BYOL, and MoCo-v3) use a fixed 2-layer predictor. For methods that incorporate a temperature parameter (e.g., SDMI, MoCo-v3, JMI, and SimCLR), an additional dimension is included in the search space. Momentum-based models such as BYOL and MoCo-v3 use a fixed

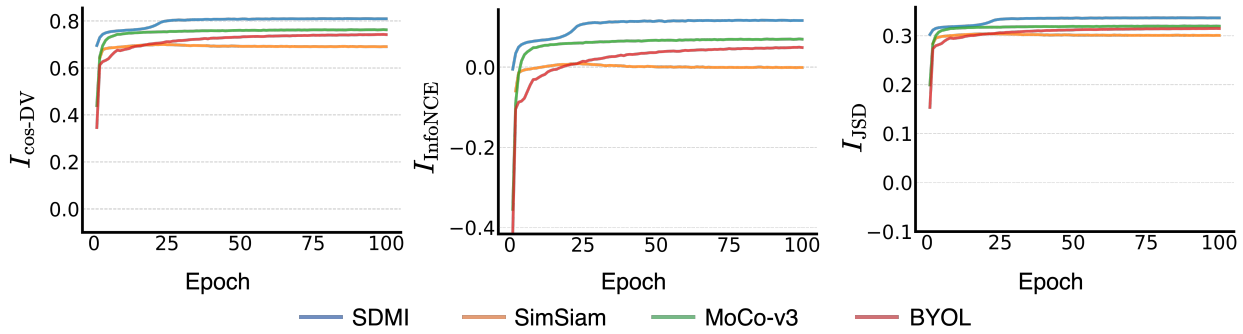


Figure 7: Estimated MI under mini-batch SGD. All methods continue to show monotonic MI growth across three estimators ($I_{\text{cos-DV}}$, I_{InfoNCE}, I_{JSD}).

momentum coefficient of 0.996 for the target encoder. The total number of configurations evaluated per model is shown in Table 4.

Hyperparameter sweep 2. Based on the results from sweep 1, we perform a secondary evaluation for each model on each dataset using 3 projection layers. The top-performing configuration for each model (shown in Table 5) is then selected and used to train that model for 1000 epochs.

Table 3: Sweep 1: hyperparameter settings and search space used in our grid search.

Parameter	**Values**	**Applies to**	**Fixed/Varied**
Encoder Backbone	ResNet-18	All models/datasets	Fixed
Batch Size	512	All models/datasets	Fixed
Projection Layers	2	All models/datasets	Fixed
Prediction Layers	2	All models/datasets	Fixed
Prediction Dimension	256	All models/datasets	Fixed
Epochs	300	All models/datasets	Fixed
Feature Dimension	2048	All models/datasets	Fixed
Momentum Coefficient	0.996	BYOL, MoCo-v3	Fixed
Seed	7349	All models/datasets	Fixed
Learning Rate	{0.01, 0.03, 0.05}	All models (cosine decay)	Varied
Weight Decay	{0.0001, 0.0005}	All models	Varied
Temperature	{0.05, 0.07, 0.1}	SDMI, JMI, MoCo-v3, SimCLR	Varied
Projection Dimension	{128, 256}	All models	Varied

Table 4: Grid search configuration counts per model for sweep 1. Configurations are counted per dataset unless otherwise noted.

Model	**Dataset(s)**	**Temperature Used**	**Total Configurations**
SDMI	CIFAR10, CIFAR100	Yes	$3 \times 2 \times 3 \times 2 = 36$
SimSiam	CIFAR10, CIFAR100	No	$3 \times 2 \times 2 = 12$
BYOL	CIFAR10, CIFAR100	No	$3 \times 2 \times 2 = 12$
MoCo-v3	CIFAR10, CIFAR100	Yes	$3 \times 2 \times 3 \times 2 = 36$
JMI	CIFAR10, CIFAR100	Yes	$3 \times 2 \times 3 \times 2 = 36$
SimCLR	CIFAR10, CIFAR100	Yes	$3 \times 2 \times 3 \times 2 = 36$
Barlow Twins	CIFAR10, CIFAR100	No	$3 \times 2 \times 2 = 12$
VICReg	CIFAR10, CIFAR100	No	$3 \times 2 \times 2 = 12$

Table 5: Optimal hyperparameters selected from sweep 2 for each model and dataset. LR = learning rate, WD = weight decay, Temp. = temperature. All models use a ResNet-18 encoder. For VICReg, we fix the similarity, variance, and covariance loss coefficients to 25.0, 25.0, and 1.0, respectively, and set a small numerical stability term $\epsilon = 10^{-4}$. These values remain constant across all runs.

Model	**Dataset**	**LR**	**WD**	**Temp.**	**Proj. Dim**	**Proj. Layers**	**Predictor**
SDMI	CIFAR10	0.05	0.0005	0.1	256	3	No
SDMI	CIFAR100	0.03	0.0005	0.1	256	2	No
SimSiam	CIFAR10	0.05	0.0005	–	256	3	Yes
SimSiam	CIFAR100	0.05	0.0005	–	256	3	Yes
BYOL	CIFAR10	0.05	0.0005	–	256	3	Yes
BYOL	CIFAR100	0.05	0.0005	–	256	2	Yes
MoCo-v3	CIFAR10	0.05	0.0005	0.1	256	2	Yes
MoCo-v3	CIFAR100	0.05	0.0001	0.1	256	2	Yes
JMI	CIFAR10	0.05	0.0005	0.1	128	2	No
JMI	CIFAR100	0.03	0.0005	0.1	256	2	No
SimCLR	CIFAR10	0.05	0.0005	0.1	128	2	No
SimCLR	CIFAR100	0.03	0.0001	0.1	128	2	No
Barlow Twins	CIFAR10	0.05	0.0005	–	256	2	No
Barlow Twins	CIFAR100	0.05	0.0005	–	256	2	No
VICReg	CIFAR10	0.05	0.0005	–	256	3	No
VICReg	CIFAR100	0.03	0.0005	–	256	3	No

C.3 ImageNet100 and TinyImageNet training configuration

For both the ImageNet100 and TinyImageNet datasets, we adopt a uniform configuration across all models, with only a few dataset-specific adjustments. The settings are summarized in Table 6.

Table 6: Hyperparameter settings used for all models trained on ImageNet100 and TinyImageNet.

Parameter	**ImageNet100**	**TinyImageNet**
Encoder Backbone	ResNet-50	ResNet-18
Epochs	800	1000
Warmup Epochs	10	5
Batch Size	256	512
Initial Learning Rate	0.05	0.05
Learning Rate Schedule	Cosine decay	Cosine decay
Weight Decay	0.0001	0.0005
Projection Layers	3	3
Projection Dimension	256	256
Feature Dimension	512	512

Note. For the SDMI model on ImageNet100, the batch size is reduced to 64 (instead of 256) because of the increased memory requirements from the two-encoder setup.

D Method classification under the SDMI/JMI taxonomy

Table 7 classifies representative SSRL methods under the SDMI/JMI framework developed in Section 3, categorizing each method by its update rule (EM-style alternating vs. joint) and by whether it optimizes an explicit MI bound or an MI surrogate.

Table 7: Representative SSRL methods and their classification under the SDMI/JMI taxonomy, with objective types.

Method	**EM/Joint**	**Objective Type**	**Paradigm**
BYOL (Grill et al., 2020)	EM	MI Surrogate	SDMI
SimSiam (Chen & He, 2021)	EM	MI Surrogate	SDMI
MoCo (He et al., 2020; Chen et al., 2020b; 2021)	EM	Explicit MI	SDMI
DINO (Caron et al., 2021)	EM	MI Surrogate	SDMI
SimCLR (Chen et al., 2020a)	Joint	Explicit MI	JMI
Barlow Twins (Zbontar et al., 2021)	Joint	MI Surrogate	JMI
VICReg (Bardes et al., 2022)	Joint	MI Surrogate	JMI
W-MSE (Ermolov et al., 2021)	Joint	MI Surrogate	JMI
SwAV (Caron et al., 2020)	Joint	MI Surrogate	JMI
BGRL (Thakoor et al., 2022)	Joint	MI Surrogate	JMI

E Algorithms

Detailed algorithmic descriptions for the canonical SDMI and JMI prototypes introduced in Section 4.1 and illustrated in Figure 1 are provided below. Following common SSRL practice (Chen et al., 2020a; 2021; Grill et al., 2020; Chen & He, 2021), we adopt a symmetric loss by computing the objective over both view orderings.

E.1 SDMI canonical form training procedure

Algorithm 1: EM-style Training Procedure of the SDMI Prototype

Input: Unlabeled dataset $\mathcal{D}$, encoders f_θ, g_ξ, temperature τ, number of epochs T
Output: Trained encoder parameters θ, ξ

for $t = 1$ **to** T **do**
  `// E-Step: Update` f_θ`, freeze` g_ξ **foreach** *minibatch* $(X_1, X_2) \sim \mathcal{D}$ **do**
    $Z_E^{(1)} \leftarrow f_\theta(X_1), \quad Z_E^{(2)} \leftarrow f_\theta(X_2)$
    $Z_M^{(1)} \leftarrow g_\xi(X_1), \quad Z_M^{(2)} \leftarrow g_\xi(X_2)$
    $\hat{Z}_M^{(1)} \leftarrow \mathrm{SG}(Z_M^{(1)}), \quad \hat{Z}_M^{(2)} \leftarrow \mathrm{SG}(Z_M^{(2)})$
    $\mathcal{L}_E \leftarrow \frac{1}{2}\left[\mathrm{DV}(Z_E^{(1)}, \hat{Z}_M^{(2)}; \tau) + \mathrm{DV}(Z_E^{(2)}, \hat{Z}_M^{(1)}; \tau)\right]$
    Update θ via gradient descent on $\mathcal{L}_E$
  end
  `// M-Step: Update` g_ξ`, freeze` f_θ **foreach** *minibatch* $(X_1, X_2) \sim \mathcal{D}$ **do**
    $Z_E^{(1)} \leftarrow f_\theta(X_1), \quad Z_E^{(2)} \leftarrow f_\theta(X_2)$
    $Z_M^{(1)} \leftarrow g_\xi(X_1), \quad Z_M^{(2)} \leftarrow g_\xi(X_2)$
    $\hat{Z}_E^{(1)} \leftarrow \mathrm{SG}(Z_E^{(1)}), \quad \hat{Z}_E^{(2)} \leftarrow \mathrm{SG}(Z_E^{(2)})$
    $\mathcal{L}_M \leftarrow \frac{1}{2}\left[\mathrm{DV}(Z_M^{(1)}, \hat{Z}_E^{(2)}; \tau) + \mathrm{DV}(Z_M^{(2)}, \hat{Z}_E^{(1)}; \tau)\right]$
    Update ξ via gradient descent on $\mathcal{L}_M$
  end
end

E.2 JMI canonical form training procedure

Algorithm 2: Joint Training Procedure of the JMI Prototype

Input: Unlabeled dataset $\mathcal{D}$, encoder f_θ, temperature τ, number of epochs T

Output: Trained encoder parameters θ

```
for t = 1 to T do
    foreach minibatch (X_1, X_2) ~ D do
        Z^(1) ← f_θ(X_1),   Z^(2) ← f_θ(X_2)
        L ← 1/2 [DV(Z^(1), Z^(2); τ) + DV(Z^(2), Z^(1); τ)]
        Update θ via gradient descent on L
    end
end
```