On Assimilating Learned Views in Contrastive Learning

Anonymous authors
Paper under double-blind review

Abstract

Transformations based on domain expertise (expert transformations), such as *random-resized-crop* and *color-jitter*, have proven critical to the success of contrastive learning techniques such as SimCLR. Recently, several attempts have been made to replace such domain-specific, human-designed transformations with generated views that are learned. However for imagery data, so far none of these view generation methods has been able to outperform expert transformations. In this work, we tackle a different question: instead of replacing expert transformations with generated views, can we constructively assimilate generated views with expert transformations? We answer this question in the affirmative. To do so, we first propose an information-theoretic framework for designing view generation based on the analysis of Tian et al. (2020b) on what makes a "good" view in contrastive learning. Then, we present two simple yet effective assimilation methods that together with our view generation mechanisms improve the state-of-the-art by up to $\approx 3.5\%$ on four different datasets. Importantly, we conduct a detailed empirical study that systematically analyzes a range of view generation and assimilation methods and provides a holistic picture of the efficacy of learned views in contrastive representation learning.

1 Introduction

Contrastive learning (CL) has become a powerful tool for self-supervised representation learning. Most contrastive methods are trained using instance discrimination: pulling positive views (generated from the same image) close in the learned representation space, while pushing negative views (generated from other images) away (Dosovitskiy et al., 2014; Wu et al., 2018; Chen et al., 2020b;c; He et al., 2020; Chen et al., 2020d; Caron et al., 2020; Grill et al., 2020; Chen & He, 2021; Caron et al., 2021; Kotar et al., 2021; Tejankar et al., 2021; Wang et al., 2021; Zbontar et al., 2021). Intuitively, the quality of representations learned by CL is highly dependent on the mechanisms used to generate these views, which are commonly compositions of a set of handcrafted transformations designed by human experts. The choice of transformations and the specific design of their composition relies on domain expertise built by years of CL research - which has mostly focused on general imagery, commonly captured with consumer cameras or collected from the web. However, this domain expertise may not necessarily hold for new unseen visual domains that could be encountered by CL practitioners. As such, many recent works (Tamkin et al., 2020; Jahanian et al., 2021; Shi et al., 2022) have attempted to remedy this issue by proposing learning-based approaches to view generation. However, none of these methods have been successful in completely replacing the expert transformations while matching the performance. Therefore, a natural question arises: *How do we systematically learn to generate "meaningful" views and, more importantly, assimilate them alongside expert views to improve CL performance?*

Recently, Tian et al. (2020a;b) proposed an information theoretic definition of optimal positive views in CL. They show that there exist a sweet spot at which the mutual information gap between the two views captures the optimal amount of task-relevant information, with all else discarded. In this work, we build upon their definition to introduce an information theoretic view generation framework. To this end, we extend the results of Zimmermann et al. (2021), who showed that contrastive learning inverts the true data generating process. We present two new results that allow us to efficiently estimate the mutual information gap between views. These new findings allow us to propose two practical view generation methods that create information-theoretically "meaningful"

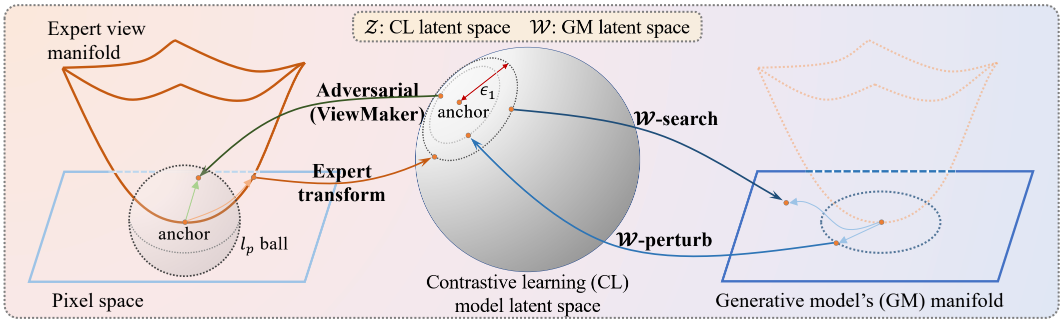


Figure 1: Illustrative visualization of the view generation strategies. The convex shaped surface represents the expert view manifold, and the plane on the left and right represent the input pixel space and the image manifold learned by a GAN, respectively. The hypersphere in the middle represents the learned representation space of a CL method. ***$\mathcal{W}$-search*** searches in the CL model's latent space ($\mathcal{W}$-space) using the loss in Eq 3 and then projects it back to the image manifold of the GAN to generate the view. ***$\mathcal{W}$-perturb*** directly perturbs the GAN's latent space ($\mathcal{W}$-space) with Gaussian noise and then pushes it through to the GAN's image manifold. **ViewMaker** (Tamkin et al., 2020) performs adversarial training and constrains the generated view in an l_p ball around the anchor in pixel space. **Expert transform** generates views by applying a set of transformations commonly used in CL model training. Note that only *$\mathcal{W}$-search* explicitly controls the distance from anchor in $\mathcal{W}$ space.

views by using pretrained generative models. Given this formal view generation framework and two practical generation strategies, we introduce two view assimilation methods to leverage the generated views along with expert views to improve performance and provide extensive empirical analysis to benchmark their performance. To the best of our knowledge, we are the first to show that GAN-generated views, when assimilated properly, can lead to consistent improvements in CL performance on four standard evaluation benchmarks.

To summarize, our main contributions are the following: (1) we propose an information-theoretic framework for view generation in CL that readily encompasses previously proposed view generation methods; (2) we instantiate two new practical methods under this framework that leverage a pretrained generative model to generate meaningful positive views; (3) we introduce a new loss that effectively assimilates the generated views into the existing InfoNCE loss; (4) with an extensive set of experiments, we show 0.9%, 2.4%, 3.5%, and 2.5% improvements above a popular CL baseline on the CIFAR10, CIFAR100, TinyImageNet, and ImageNet datasets respectively, and demonstrate significantly faster convergence; (5) we provide thorough ablation studies and empirical analysis of other view generation alternatives for CL (see Fig. 1 for a high-level summary).

2 Background and Related Work

SimCLR (Chen et al., 2020b;c) is one of the first and most established CL baselines. Let $x \in \mathcal{X}$ denote an image in a mini-batch $\mathcal{B} = \{x_i\}_{i=1}^N$. Further, let $f : \mathcal{X} \mapsto \mathcal{W}$ be a representation encoder, parameterized as a deep neural network. Here $\mathcal{X} \subseteq \mathbb{R}^D, \mathcal{W} \subseteq \mathbb{R}^K$ such that $K < D$. Following the notations from Khosla et al. (2020), we use $i \in \mathcal{I} \equiv \{1, \ldots, 2N\}$ to denote the index of an arbitrary batch (augmented using *expert transformations* as explained below), where $j(i)$ is the index of the other augmented sample originating from the same data sample. $\mathcal{A}(i) := \mathcal{I} \backslash \{i\}$ is the complement of i. The contrastive loss function for SimCLR is the InfoNCE loss, which is

$$L_{\text{simclr}} = -\sum_{i \in \mathcal{I}} \log \frac{\exp(z_i \cdot z_{j(i)}/\tau)}{\sum_{a \in \mathcal{A}(i)} \exp(z_i \cdot z_a/\tau)} \tag{1}$$

where, $z = f(x)$ and τ is the temperature parameter.

Views In contrastive learning, a stochastic data augmentation module applies two different sets of transformations to generate two correlated views of the same data point, x_i and $x_{j(i)}$. The two sets of transformations, t_1 and t_2, are sampled from the same family of transformations $\mathcal{T}$, such as cropping and resizing, horizontal flipping, color distortion, etc. The images x_i and $x_{j(i)}$ are the positive views, whereas all other views generated from other samples are the negative views. As studied by many recent works (Ye et al., 2019; Misra & van der Maaten, 2020; Tian et al., 2020a), $\mathcal{T}$ defines the invariances that the model learns. Therefore, human domain knowledge and extensive years of research have been invested to search for the optimal composition of augmentations and their corresponding parameters to optimize CL methods for a given data. We refer to the optimal set of transformations in SimCLR as *expert transformations* and views created by them as *expert views*.

View Generation Previous works have explored the idea of generating views for SSL. ADIOS (Shi et al., 2022) and Viewmaker (Tamkin et al., 2020) both study adversarial methods. ADIOS learns a masking function and an image encoder performing a min-max optimization on the same objective function in the masked image model framework. Alternatively, Viewmaker learns a bounded perturbation directly in the pixel space by employing the min-max adversarial training. Relatedly, Jahanian et al. (2021) leverages a pre-trained GAN model to generate the data samples and to replace the entire training dataset like ImageNet Deng et al. (2009) with a generated one.

View Assimilation Previous works like CMC Tian et al. (2020a) and DINO Caron et al. (2021) study the case of having more than two views in contrastive learning. Both works treat this as a special case in which an arbitrary batch of views has been expanded and demonstrate gains from including additional views. However, empirically we observed that simply augmenting SimCLR with an additional set of positive views degrades its performance. Instead, we explore and propose new methods that can effectively integrate additional views and boost performance.

3 View Generation

Tian et al. (2020a;b) introduced an information theoretic definition of what makes for a "good" positive view in contrastive learning. Letting $I(\cdot;\cdot)$ be the mutual information (MI), they state the following proposition:

Proposition 3.1. *Suppose f is a minimal sufficient encoder. Given a downstream task T with label y, the optimal views from the data x are $(x_i^*, x_{j(i)}^*) = \arg\min_{x_i, x_{j(i)}} I(x_i; x_{j(i)})$, subject to $I(z_i; y) = I(z_{j(i)}; y) = I(x; y)$. Given $x_i^*, x_{j(i)}^*$, the representations z_i^* (or $z_{j(i)}^*$) learned by contrastive learning is optimal for T, thanks to the minimality and sufficiency of f.*

In other words, there exists an optimal level of mutual information gap between the positive views such that only the task-relevant information is preserved with all others discarded during training, i.e. $I(z_i; z_{j(i)}) = I(x_i; y_i)$. In this section, we will use this definition of optimal views to prescribe a view generation mechanism. A successful generation mechanism requires solving the following two challenges: (1) estimating mutual information in a high-dimensional observation space; and (2) evaluating the amount of task-relevant information contained in the views without access to labeled data from the downstream task in SSL.

Estimating mutual information To leverage Proposition 3.1, one needs to be able to estimate $I(z_i; z_{j(i)})$ and potentially $I(x_i; y_i)$. However, owing to the high-dimensionality of these spaces, it is not possible to estimate these quantities accurately. Thus to resolve this, we present a new result based on the recent result of Zimmermann et al. (2021) that states the following proposition:

Proposition 3.2. *(from Zimmermann et al. (2021)) Let $\mathcal{W}$ be a unit hypersphere $\mathbb{S}^{K-1}$, $\tau > 0$ and consider the ground-truth conditional distribution of the form $p(\tilde{w}|w) = C_p^{-1} e^{\kappa \tilde{w}^\top w}$. Let $h = f \circ g$, where $g : \mathcal{W} \mapsto \mathcal{X}$ is a generator, map onto a hypersphere with radius $\sqrt{\tau\kappa}$. Consider the conditional distribution q_h parameterized by the model, $q_h(\tilde{w}|w) = C_h(w)^{-1} e^{h(\tilde{w})^\top h(w)/\tau}$ with $C_h(w) := \int e^{h(\tilde{w})^\top h(w)/\tau} d\tilde{w}$, where the hypothesis class for h (and thus f) is assumed to be sufficiently flexible such that $p(\tilde{w}|w)$ and $q_h(\tilde{w}|w)$ can match. If $h^\star$ is a minimizer of the cross-entropy $\mathbb{E}_{p(\tilde{w}|w)}[-\log q_h(\tilde{w}|w)]$, then $q_{h^\star}(\tilde{w}|w) = p(\tilde{w}|w)$ and $\forall w, \tilde{w} : \tau\kappa w^\top \tilde{w} = h^\star(w)^\top h^\star(\tilde{w})$.*

Simply put, Proposition 3.2 implies that contrastive learning, given a sufficiently flexible f, inverts the true (unknown) data generating process g. We now state our following proposition for efficiently estimating the mutual information (MI):

Proposition 3.3. *Let* $\mathcal{W} = \mathbb{S}^{K-1}$, $p(z) = |\mathcal{W}|^{-1}$, *and conditional* $p(\tilde{z}|z)$ *is a von Mises-Fisher (vMF) distribution:*

$$p(\tilde{z}|z) = C_p^{-1} e^{\tilde{z}^\top z/\tau}, \quad \textit{with} \quad C_p := \int e^{\tilde{z}^\top z/\tau} d\tilde{z}. \tag{2}$$

Then the mutual information $I(\tilde{Z}; Z) = \frac{1}{\tau}\mathbb{E}_{z,\tilde{z}}\left[z^\top \tilde{z}\right] + \textit{const.}$

The proof is in Appendix C. Proposition 3.3 states that we can directly use the distances in the CL latent space to estimate the mutual information between views up to a scaling and shift operation. Since the latent space in CL is normalized to a unit hypersphere, the inner-products are equivalent to the L2 (Euclidean) distances. In Table 1 and Figure 2, we empirically demonstrate the strong correlation between the MI between views and their Euclidean distances in the latent space, by comparing against the MI estimates from MINE (Belghazi et al., 2018), which are theoretically a lower bound to the true MI. In Table 1, the MI estimates from MINE and the L2 distances between views are given by the first and second columns, respectively. The strong correlation between the two MI measures in this table and the corresponding illustrative Figure 2 empirically confirms our Proposition 3.3.

Lack of access to labeled data To tackle the problem of solving for the optimal amount of task-relevant information without access to the task T, since the latent space is normalized, we can indeed conduct a grid search to find the optimal mutual information gap. However, through extensive empirical studies, previous works have proven that expert transformations create a reasonably sufficient amount of MI gap. Leveraging on this insight, as well as the correlation between MI and the Euclidean distances between view representations derived in Proposition 3.3, a good heuristic we adopt is to use the expectation of the euclidean distances between the anchor and the expert views.

Table 1: Estimated mutual information (using the MINE Belghazi et al. (2018)) and the average L2 distances of CL encoder latents between the anchor (original), positive expert view (expert) and generated views.

	$I(f(X); f(\tilde{X}))$		$\mathbb{E}[f(x)^\top f(\tilde{x})]$	
View Pairs	CIFAR10	CIFAR100	CIFAR10	CIFAR100
Original, Expert	4.14	5.41	0.973	0.986
Original, $\mathcal{W}$-*search*	4.13	4.40	0.907	0.867
$\mathcal{W}$-*search*, Expert	3.78	4.35	0.888	0.859
Original, $\mathcal{W}$-*perturb*	3.79	3.91	0.849	0.796
$\mathcal{W}$-*perturb*, Expert	3.66	3.86	0.832	0.789

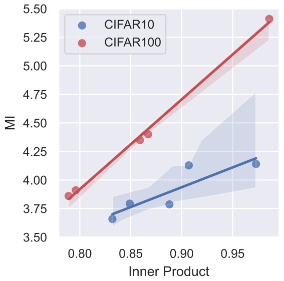


Table 2: Inner-Products v.s. Mutual Information estimates from MINE.

View generation framework We can now cast view generation as a problem of searching for the $w^\star$ that minimizes the following loss function. Assuming we are generating n views simultaneously,

$$\{w_k^\star\}_{k=1}^n = \underset{\{w_k\}}{\arg\min} \left\{ \frac{1}{n} \sum_k \underbrace{\delta\left(\epsilon_1, \|f \circ g(w_k) - f(x_0)\|_2\right)}_{\text{boundary constraint}} + \underbrace{\lambda(\epsilon_2 - \bar{d}_n)^+}_{\text{uniformity}} \right\}, \tag{3}$$

where $\delta(a, b)$ is the L2 loss, i.e. $\delta(a, b) = (a - b)^2$, $\bar{d}_n$ is the average Euclidean distances among generated views, i.e. $\bar{d}_n = \frac{1}{n(n-1)} \sum_{j \neq k} \|f \circ g(w_j) - f \circ g(w_k)\|_2$, and $(\cdot)^+ := \max(0, \cdot)$ is a ReLU function. The first term is a boundary constraint which enforces the generated views to be ϵ_1 away from the anchor in the representation space. The second term is a uniformity regularization that forces the generated views to be spread out (at least ϵ_2 away on average) and therefore diverse. A hinge loss is utilized in this regularization term since we do not need to precisely control the pairwise distances among the views. Both ϵ_1 and ϵ_2 are hyperparameters. Note that Eq. (3) is an exact realization of Proposition 3.3, but assumes prior access to the true data generating process g. In practice, we need to approximate g by a generative model of the true data generation process. This leads to our first view generation method that we describe next.

3.1 $\mathcal{W}$-SEARCH: PERTURBATION IN CL ENCODER'S LATENT SPACE

We propose $\mathcal{W}$*-search* as a straightforward implementation of Eq. (3) using a pretrained generator g. Under the assumption that we can learn a generative model g and encoder e to approximate the true data generating process (i.e., the joint distribution of (x, w) matches, $p_{g,e} = p_{x,w}$)[*], Proposition 3.3 can be strictly realized.

Practical implementation In practice, we further relax this assumption by using a standard GAN generator without jointly training an inverter. But, we utilize GAN inversion to initialize w for the optimization. In our experiments, we use a pretrained StyleGAN (Karras et al., 2019; 2020b) as g and In-domain GAN Inversion (Zhu et al., 2020) as e.

Finally, we define $\mathcal{W}$*-search* as a transformation $\mathcal{W}$*-search*$(x) := g(w^\star)$ where $w^\star$ is the minimizer of the loss defined in Eq. (3). $\mathcal{W}$*-search* is illustrated in Fig. 1. Intuitively, it is designed to generate additional positive views whose *distribution of distances* from the anchor image x in the CL encoder f latent space is similar to that of the expert views. Examples of generated views are given in Fig. 3. Note that $\mathcal{W}$*-search* differs significantly from ViewMaker (Tamkin et al., 2020) in that $\mathcal{W}$*-search* is not formulated as an adversarial game against the same InfoNCE objective that the contrastive encoder learns to minimize.

Scalability of $\mathcal{W}$*-search* While effective, online $\mathcal{W}$*-search* is computationally expensive because the optimization involving both the contrastive encoder and the generative model needs to be performed for every image in the mini-batch. Therefore, this online view generation using $\mathcal{W}$*-search* does not scale as well to large-scale datasets. This scalability problem can be solved by performing view generation *offline*. By leveraging a pretrained f, we can cache the generated views before the actual CL training. In the following experiments, we focus on this offline setting via caching, and provide an empirical study of approximated online version in Appendix E.

3.2 $\mathcal{W}$-PERTURB: PERTURBATION IN GM'S LATENT SPACE

Another solution to the scalability issue of the online $\mathcal{W}$*-search* is to generate views via perturbations in the latent space $\mathcal{W}$ of the generator, instead of the latent space of the contrastive encoder $\mathcal{Z}$. Thereby, we remove the computationally expensive step of finding views through optimization. In fact, when assumptions in Zimmermann et al. (2021) are realized and the latent spaces of the generator and CL encoder are well-aligned, the mapping $h = f \circ g : \mathcal{W} \mapsto \mathcal{Z}$ becomes linear and the two latent spaces are coupled by a rotation matrix. Given that MI is invariant under rotation, from Proposition 3.3 we can in turn relate the mutual information between views to the inner-products between w's.

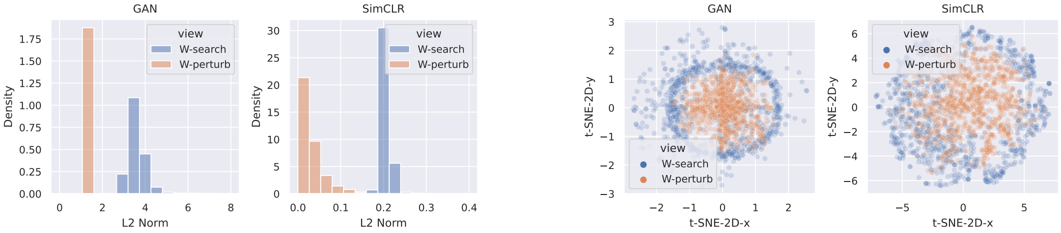
Corollary 3.3.1. *Assuming the ground-truth data generating process is g. When Theorem 2 from Zimmermann et al. (2021) holds, f recovers the latent sources $\mathcal{W}$ up to an orthogonal linear transformation and a constant scaling factor. Let $\mathcal{W} = \mathbb{S}^{K-1}$, $p(w) = |\mathcal{W}|^{-1}$, and $p(\tilde{w}|w)$ is a vMF distribution, $p(\tilde{w}|w) = C_p^{-1}\exp(\kappa\tilde{w}^\top w)$. Let $h := f \circ g$, then with the optimal $h^\star$ that solves $\lim_{N\to\infty} L_{simclr}$ 1, the mutual information $I(Z;\tilde{Z}) = I(h^\star(W); h^\star(\tilde{W})) = \kappa\mathbb{E}_{w,\tilde{w}}\left[w^\top\tilde{w}\right] + const.$*

This corollary allows us to propose an alternative view generation method, $\mathcal{W}$*-perturb*, that creates positive views by directly perturbing in the latent space $\mathcal{W}$ of the pretrained generator g. Under this method, additional positive views for a given anchor x are generated as: $\tilde{x} = \mathcal{W}$*-perturb*$(x) := g(e(x) + w_p)$, where $w_p \sim \mathcal{N}(0, \sigma I)$[†] and $e(x)$ is the projection of the anchor image in the latent space of g. This is a generalization of the *latent transforms* $T_{\mathbf{z}}$ introduced in Jahanian et al. (2021). The latent transform is not directly applicable to real image domain since the corresponding latents are unknown. Thus, we project real images in GAN's latent space via its inverter e.

When all assumptions are realized, $\mathcal{W}$*-perturb* is exactly equivalent to $\mathcal{W}$*-search*. However, if in practice $\mathcal{W}$ and $\mathcal{Z}$ are misaligned, $\mathcal{W}$*-perturb*'ed views will not be equally distant from the anchor in

[*] For example, training a bidirectional GAN (Donahue et al., 2016; Srivastava et al., 2017) minimizes the Jensen-Shannon divergence, $JS(p_{x,w}||p_{g^\star,e^\star}) = 0$

[†] In practice, $\mathcal{W}$ and $\mathcal{Z}$ do not need to have the same dimension. Assuming $w \in \mathbb{R}^{K'}$, we can project w_p on to a hypersphere with radius $r = \sigma\sqrt{K'}$. Please see Tab. 16 for a comparison.



(a) L2 norms

(b) t-SNE

Figure 2: Visualization of GAN and SimCLR latent spaces in CIFAR100. (a) L2 norms between anchors and generated views. We can see $\mathcal{W}$*-search* induces a much tighter distribution around the target ϵ_1 value. (b) 2-D t-SNE plots of $\mathcal{W}$*-search* and $\mathcal{W}$*-perturb*. For the GAN's latent space, t-SNE's perplexity is 40. For SimCLR's latent space, t-SNE's perplexity is 20. The hyperparameters for view generation are the following: $\epsilon_1 = 0.2$, $\epsilon_2 = 0.35$, and $\sigma = 0.2$.

$\mathcal{W}$. By design, $\mathcal{W}$*-search* explicitly controls the distances in $\mathcal{Z}$. Additionally, the analysis in Fig. 2 illustrates the empirically observed misalignment between $\mathcal{W}$ and $\mathcal{Z}$. We observe that $\mathcal{W}$*-perturb* results in a one-sided heavy-tail distribution in the SimCLR model latent space, whereas $\mathcal{W}$*-search* leads to view representations that are more concentrated on a ring around the anchor. The fact that the two view generation methods are not precisely aligned suggest that, in practice, since we do not have the true data generating process and only approximate it, the two latent spaces of $\mathcal{W}$ and $\mathcal{Z}$ may not be perfectly aligned.

4 View Assimilation

In the case of SimCLR, prior works (Tamkin et al., 2020) have explored replacing expert views entirely with generated views. However, a complete replacement leads to a degradation in performance. Taking the MI maximization perspective of the InfoNCE loss (Wu et al., 2020; Tian et al., 2020b; Van den Oord et al., 2018; Poole et al., 2019) in SimCLR, a possible explanation for the degradation in performance can be attributed to the differences in the MI between the anchor and expert views and the MI between the anchor and generated views. While it is difficult to accurately compute MI in the high-dimensional $\mathcal{X}$ space, in Table 1, we provide estimates for preliminary analysis. We find that while the original and expert views share roughly the same amount of mutual information as the original and generated views, the shared information seem to be different given that there is a similar gap in information between the expert and the generated views. This finding indicates that the generated views are likely to contain meaningfully **complementary** information to the expert views and hence could lead to additional useful features (for the downstream task). Altogether, these observations motivate us to assimilate generated views into contrastive learning, instead of entirely replacing the expert views, to improve downstream accuracy. To this end, we propose two methods for *assimilating* generated views into CL training.

Replacement (A1) Our first assimilation method replaces only one of the two expert views with a generated view. On the generated view, we apply a weak amount of *random-resized-crop* and *flipping*.

Multiview (A2) Our second assimilation method simply casts the problem as multiview contrastive learning, where there are more than two positive views. We append the additional positive views, $x_{k(i)}$, and define $\{k(i)\}$ as the batch indices of the appended view(s) generated from a anchor image with the index i. This, however, requires an adjustment in the training loss. To this end, we propose the following multiview loss:

$$L_{\text{multiview}} = L_{\text{CL}} - L_{\text{align}}, \quad \text{where} \quad L_{\text{CL}} = L_{\text{InfoNCE}} \quad \text{and} \quad L_{\text{align}} = \sum_{i \in \mathcal{I}} \frac{\alpha}{|k(i)|} \sum_{p \in k(i)} z_i^\top z_p / \tau \tag{4}$$

Our loss function appends an α-weighted sum of dot products of the projected (to the CL embedding space) anchor view i and its respective generated views $k(i)$ to the base CL loss L_{CL}. L_{align} is a general plug-in term that can be used in conjunction with other existing contrastive losses. However, we empirically found it to work best with InfoNCE (Van den Oord et al., 2018) and adopt it as our L_{CL} unless otherwise specified. Please refer to Appendix F for results where we use the loss from SimSiam (Chen & He, 2021). We also experimented with the multiview loss from SupCon (Khosla et al., 2020), but found it to perform more poorly than our proposed loss.

5 Empirical Study

In this section, we conduct a comprehensive empirical study of view generation and assimilation methods for CL and provide a thorough benchmark of performances. For this purpose, we use the highly optimized SimCLR implementation from Dangovski et al. (2021) for all examined methods and benchmark them on their downstream classification accuracy on four datasets: CIFAR10, CIFAR100, TinyImageNet and ImageNet. We report the linear probing accuracy as the main evaluation metric. Additional experimental details and hyperparameters are provided in Appendix A. We experiment primarily with SimCLR because alignment and uniformity have only been formally studied for the InfoNCE loss family (Zimmermann et al., 2021; Wang & Isola, 2020). However, our view generation and assimilation methods also work with other SSL approaches like SimSiam (see Appendix F).

5.1 View Generation and Assimilation

We start with our main results, ablating all possible combinations of four different view generation methods and two view assimilation methods. For the view generation methods, we use our proposed $\mathcal{W}$*-search* and $\mathcal{W}$*-perturb* methods along with Viewmaker (Tamkin et al., 2020) and the expert transformations from SimCLR (Chen et al., 2020a). For assimilation of generated views, we consider replacement of one expert view (A1) and multiview (A2) as described in Sec. 4. For evaluation, we report the top-1 linear probe accuracy (denoted as Acc@1) and k-Nearest-Neighbor accuracy ($k = 5$, denoted as 5-NN). To obtain linear probe accuracy, we freeze the backbone of f and train a linear layer with SGD for 100 epochs. To determine the value of ϵ's for $\mathcal{W}$*-search*, we first pretrain a SimCLR encoder using expert views and compute the ϵ as the average distance (in $\mathcal{W}$-space) between anchors and their expertly transformed views. For $\mathcal{W}$*-perturb*, we conduct a grid search on the σ. Further details on these hyperparameters are in Appendix G.

As shown in Tab. 3, both $\mathcal{W}$*-search* and $\mathcal{W}$*-perturb* outperform all other view generation methods on CIFAR10, CIFAR100, and TinyImageNet. When we replace one of the views with our proposed view generation strategies, except in CIFAR10, we see consistent improvements and $\mathcal{W}$*-search* proves to be a more effective generation method. Especially for TinyImageNet, we see an improvement of $\sim 4\%$. When we augment the generated views for multiview contrastive learning, in contrary to the intuition that more expert views should improve performance, assimilating a third expert view in fact degrades performance in most cases. On the other hand, the views we generate with $\mathcal{W}$*-search* and $\mathcal{W}$*-perturb* consistently lead to improvements of $0.9\%, 2.3\%$, and 3.6% on CIFAR10, CIFAR100 and TinyImageNet respectively. Overall, the fact that our generated views, when replacing one view or being assimilated, almost always leads to an improvement suggests that our information-theoretic framework for view generation allows for generating views that capture some different information from the expert transformations for the downstream task.

We also evaluate against the standard large-scale dataset ImageNet. Following the more difficult setting in Jahanian et al. (2021), we generate 1.3 million "fake" images using BigBiGAN (Donahue & Simonyan, 2019) (approximately the number of images in ImageNet) from the GAN and only train on this generated dataset. However, we follow the standard protocol of reporting the linear probe accuracy on the real ImageNet dataset. As evident in Table 4, adding our generated view as an additional positive view improves performance by $\sim 1.5\%$ and proves to capture some meaningful information that expert transformations do not.

5.2 Replacing Expert Transformation

We now present a comprehensive study on the impact view generation methods, assimilation methods, and different losses have on the complete replacement of the expert views in SimCLR. We define

Table 3: Linear probe accuracy for the four view generation methods ($\mathcal{W}$*-search*, $\mathcal{W}$*-perturb*, Viewmaker, expert transformation) under A1 and A2 view assimilation methods. We also report the baseline SimCLR (reproduced) and Viewmaker (reproduced) accuracies in rows 1 and 5.

View 1	View 2	View 3	Loss	CIFAR10	CIFAR100	TinyImageNet
expert	expert	✗	SimCLR	92.04	70.41	47.48
expert	$\mathcal{W}$*-search*	✗	A1	91.86	71.69	**51.08**
expert	$\mathcal{W}$*-perturb*	✗	A1	91.09	70.83	50.18
expert	ViewMaker	✗	A1	82.91	41.87	26.40
ViewMaker	ViewMaker	✗	SimCLR	83.59	44.04	40.53
expert	expert	expert	A2	91.46	70.76	47.19
expert	expert	$\mathcal{W}$*-search*	A2	**92.90**	72.76	51.05
expert	expert	$\mathcal{W}$*-perturb*	A2	92.38	**72.95**	50.73
expert	expert	ViewMaker	A2	80.07	36.51	25.30

Table 4: ImageNet experiments. Linear probe accuracies are reported.

View 1	View 2	View 3	Loss	Acc@1
expert	expert	✗	SimCLR	49.93
expert	$\mathcal{W}$*-perturb*	✗	A1	48.81
expert	expert	$\mathcal{W}$*-perturb*	A2	**51.42**

the *basic* transform as random-resized-crop with the same hyperparameters as in expert transforms (for CIFAR100, area range 0.2-1, aspect ratio range 3/4-4/3). Cropping `crop`, `crop`, `crop` are defined with aspect ratio range 0.9-1.1, and area range of 0.5-1, 0.7-1, 0.9-1, respectively. Horizontal flipping is always added. We conclude from Table 5 that, while replacing both expert views with generated views under-performs, $\mathcal{W}$*-search+basic*, using our A2 loss gets very close to the baseline accuracy (panel (b) row 3, 69.55%). We found that $\mathcal{W}$*-search* is substantially better than $\mathcal{W}$*-search* when replacing both of the expert views, as shown in Table 5(a). However, applying a small amount of *random-resized-cropping* is not only necessary to prevent overfitting (as the model can eventually (after 600 epochs) memorizes the cached images) but also improves performance, as shown.

Table 5: Ablation on different configurations on complete replacement of expert views in the CIFAR100 dataset.

(a)

View 1	View 2	Acc@1	5-NN
$\mathcal{W}$*-search*	$\mathcal{W}$*-search*	57.65	50.61
$\mathcal{W}$*-search*+`crop`	$\mathcal{W}$*-search*+`crop`	62.93	54.38
$\mathcal{W}$*-search*+`crop`	$\mathcal{W}$*-search*+`crop`	66.47	58.72
$\mathcal{W}$*-search*+`crop`	$\mathcal{W}$*-search*+`crop`	66.64	59.44
$\mathcal{W}$*-search*+`basic`	$\mathcal{W}$*-search*+`basic`	65.91	58.38
$\mathcal{W}$*-perturb*+`crop`	$\mathcal{W}$*-perturb*+`crop`	55.87	44.74

(b)

View 1	View 2	View 3	Loss	Acc@1	5-NN
basic	basic	✗	SimCLR	47.61	35.86
basic	$\mathcal{W}$*-search*	✗	SimCLR	69.19	62.36
basic	basic	$\mathcal{W}$*-search*	A2	**69.55**	**63.65**
$\mathcal{W}$*-search*	$\mathcal{W}$*-search*	basic	A2	67.65	60.84
basic	$\mathcal{W}$*-search*	$\mathcal{W}$*-search*	A2	69.05	62.33

5.3 Ablation Studies

Training loss The overall performance of generated-view assimilation also depends heavily on the training loss. To demonstrate that we conduct an ablation of training losses, as reported in Tab. 11. We found A2 with $\alpha = 0.5$ to be clearly better on both CIFAR10 and CIFAR100 and as such it is not order-agnostic. From Figure 4, we can see that our models that use our proposed view generation and assimilation strategies exhibit faster convergence than the baseline SimCLR model.

Caching v.s. on-the-fly and number of views For all the experiments, we create a cached set of 8 generated views per anchor and randomly sample only 1 per anchor for each training iteration. In this section, we answer two important questions: (1) What is the impact of caching $n = 8$ views instead of generating them on-the-fly? (2) How does changing the number of positive views during training influence performance?

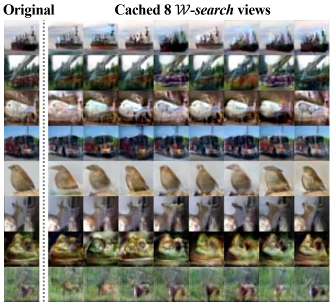


Figure 3: Visual samples of *$\mathcal{W}$-search* for CIFAR10 dataset. For each row, we show 8 cached views ($\epsilon_1 = 0.3$).

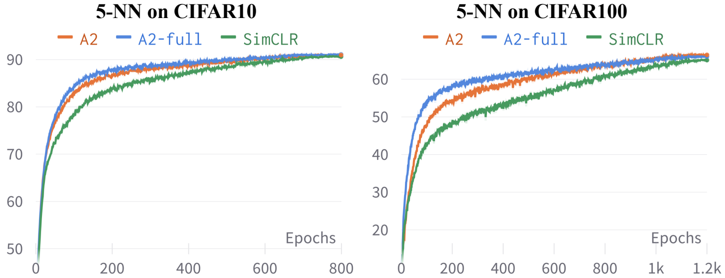


Figure 4: 5-NN accuracy curves during training. Models are trained with *$\mathcal{W}$-search* as the view generation and multiview (A2) as the loss. It is clear that the convergence speed of our models is much faster than that of the baseline SimCLR.

We investigate the first question of caching vs. on-the-fly view generation with *$\mathcal{W}$-perturb* and expert transformations in Table 6. As explained above, for *$\mathcal{W}$-search*, because the gradients from the Eq. (3) loss need to be propagated through both f and g, on-the-fly generation could be computationally expensive. Expectedly, on-the-fly view generation leads to slight improvements over the cached version. However, interestingly, the improvement from generating views on-the-fly is significantly smaller in the case where we leverage a generative model to sample the views than in the case where we use expert transformations. Although the same study could not be conducted for *$\mathcal{W}$-search* due to compute budget constraints, we conjecture that only a marginal improvement may come from using an on-the-fly generation method for *$\mathcal{W}$-search*.

To study the impact of the number of positive views during training, we conduct an ablation study with different numbers of cached views and number of views sampled during training in Table 7 with *$\mathcal{W}$-search* and A2. Contrary to the intuition that the more numbers of views we include during training the better the performance should be, we see that it is optimal to use only 1 view out of the n cached views at a time.

Table 6: Ablation on caching. Experiments are conducted on the CIFAR10 dataset. Note that the on-the-fly 3-view SimCLR baseline is equivalent to the Expert + A2 setting.

Setting	On-the-Fly?	Acc@1	5-NN
$\mathcal{W}$-perturb + A2	✗	92.38	90.53
	✓	92.64	90.76
Expert + A2	✗	91.46	90.43
	✓	92.19	90.98
3-View SimCLR	✓	91.84	90.70

Table 7: Ablation on number of cached views and number of sampled views during training. Experiments are conducted on CIFAR10.

# Cache	# View	Acc@1	5-NN
16	1	92.65	90.86
8	8	92.40	91.11
8	4	92.07	91.03
8	1	**92.90**	90.95
4	4	92.42	91.05
4	1	92.23	90.40
2	2	92.35	90.85
2	1	92.59	91.06
1	1	92.25	90.98

6 Conclusion

In this work, we introduced new findings based on recent work and an information theoretic framework for view generation. Under this framework, we proposed and presented a comprehensive study on view generation and assimilation techniques in CL. We showed that when used in conjunction with the expert views using our assimilated methods, views generated via a GAN consistently improve downstream classification performance on four different datasets.

7 Reproducibility Statement

To ensure that our reported results are reproducible, we provide an anonymized source code. In the Appendix, we provide details on the dataset, evaluation protocol, training configurations, hyperparameters, and hardware configurations. Moreover, for all findings that we derive further from previous work, we showcase the proofs for the propositions in the Appendix C. We conduct thorough and systematic ablation studies of our and previous works' methods of view generation and assimilation and present the results in both the main text and the Appendix.

References

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 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 International Conference on Computer Vision (ICCV)*, 2021.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning, (ICML)*, 2020b.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020c.

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

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

Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljačić. Equivariant contrastive learning. *arXiv preprint arXiv:2111.00899*, 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. Ieee, 2009.

Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *Advances in neural information processing systems*, 32, 2019.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, pp. 766–774, 2014.

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

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014a.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

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 (NeurIPS)*, 2020.

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)*, pp. 9729–9738, 2020.

Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. *arXiv preprint arXiv:2106.05258*, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020a.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020b.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Klemen Kotar, Gabriel Ilharco, Ludwig Schmidt, Kiana Ehsani, and Roozbeh Mottaghi. Contrasting contrastive self-supervised representation learning pipelines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9949–9959, 2021.

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

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

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6706–6716, 2020.

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

Yuge Shi, N Siddharth, Philip Torr, and Adam R Kosiorek. Adversarial masking for self-supervised learning. In *International Conference on Machine Learning*, pp. 20026–20040. PMLR, 2022.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, 30, 2017.

Alex Tamkin, Mike Wu, and Noah Goodman. Viewmaker networks: Learning views for unsupervised representation learning. *arXiv preprint arXiv:2010.07432*, 2020.

Ajinkya Tejankar, Soroush Abbasi Koohpayegani, Vipin Pillai, Paolo Favaro, and Hamed Pirsiavash. Isd: Self-supervised learning by iterative similarity distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9609–9618, 2021.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020a.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020b.

Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pp. arXiv–1807, 2018.

Guangrun Wang, Keze Wang, Guangcong Wang, Phillip HS Torr, and Liang Lin. Solving inefficiency of self-supervised representation learning. *arXiv preprint arXiv:2104.08760*, 2021.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel L. K. Yamins, and Noah D. Goodman. On mutual information in contrastive learning for visual representations. *ArXiv*, abs/2005.13149, 2020.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3733–3742, 2018.

Mang Ye, Xu Zhang, PongChi Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6203–6212, 2019.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning (ICML)*, 2021.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 33:7559–7570, 2020.

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pp. 592–608. Springer, 2020.

Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pp. 12979–12990. PMLR, 2021.

A Experimental Setup and Details

Datasets. Experiments are conducted on four datasets:

- CIFAR10 (Krizhevsky et al., 2009) has 10 classes, 50,000 images for training and 10,000 for testing.
- CIFAR100 (Krizhevsky et al., 2009) has 100 classes, 50,000 images for training and 10,000 for testing.
- TinyImageNet is introduced in Le & Yang (2015). The dataset contains 200 classes, 100,000 images for training and 10,000 for testing. Images are resized to 64×64.
- ImageNet (Deng et al., 2009) contains approximately 1.3 million images. We following the setting in Jahanian et al. (2021) and generate 1.3 million "fake" images using a pretrained BigBiGAN (Donahue & Simonyan, 2019) at resolution 128×128.

Implementation. We implement our methods based on the E-SSL Dangovski et al. (2021) codebase[‡] (for CIFAR10 experiments) and the SupCon Khosla et al. (2020) codebase[§] (for CIFAR100 and TinyImageNet). Code is provided in the Supplementary `zip` file. Hyperparameters for each dataset are listed in Table 9[¶].

TinyImageNet. For the TinyImageNet dataset, we initially tried to develop upon the codebase[‖] provided in Ermolov et al. (2021) and reproduced the SimCLR baseline results reported in their paper (see Table 8, note that $\mathcal{W}$-*search*+A2 still improves the SimCLR baseline by a large margin). However, Ermolov et al. (2021) uses a different optimizer (Adam instead of SGD as for other datasets) and a different implementation of the SimCLR loss function. Thus, for a fair comparison, we reimplement the experiments on TinyImageNet using the SupCon Khosla et al. (2020) codebase.

Viewmaker. For Viewmaker Tamkin et al. (2020), we reproduced the reported accuracy on CIFAR10. For CIFAR100 and TinyImageNet, since the original authors did not evaluate their model against these datasets, we tried our best to optimize the hyperparameters, such as optimizer, learning rate, temperature, architecture of encoder (ResNet18 small, ResNet18, ResNet50), and projection head. We describe the best set of hyperparameters in Table 10.

Evaluation. As for evaluation metrics, we adopt the conventions in the respective codebases. For CIFAR10, we run linear probe evaluations (for 100 epochs) with 5 random seeds and report the mean and standard deviation of accuracies. For CIFAR100 and TinyImageNet, we run linear probe evaluations for 100 epochs and report the best accuracy. For both settings, we load and freeze the last checkpoint of the backbone network.

Table 8: Linear probe (top-1 and top-5) and 5-NN accuracies on TinyImageNet, using an alternative implementation.

	Acc@1	Acc@5	5-NN
SimCLR	48.50	74.51	32.83
$\mathcal{W}$-*search*+A2 (ours)	**51.30**	**77.17**	**36.90**

Computational cost for view generation. The computation time depends on the hyperparameters (the number of optimization steps for $\mathcal{W}$-*search*), e.g., caching 8 views per sample for CIFAR10 takes 12.19 A100 GPU hours.

Computational cost for pretraining. Each experiment is run on 4 NVIDIA V100 GPUs. The pretraining time of SimCLR baseline for CIFAR10, CIFAR100, and TinyImagenet are 11.5, 13.6, and 29.7 hours, respectively.

[‡] https://github.com/rdangovs/essl/tree/main/cifar10

[§] https://github.com/HobbitLong/SupContrast

[¶] For CIFAR10, we found that batch size of 128 gives similar or slightly better results than the default 512.

[‖] https://github.com/htdt/self-supervised

Table 9: Hyperparameters for experiments.

	CIFAR10	CIFAR100	TinyImageNet
Optimizer	SGD	SGD	SGD
Learning Rate	0.015	0.5	0.5
Weight Decay	5e-5	1e-4	1e-4
Momentum	0.9	0.9	0.9
Cosine Decay	✓	✓	✓
Batch Size	128	512	512
SimCLR Loss	InfoNCE Eq. 12	SimCLR	SimCLR
Temperature	0.5	0.5	0.5
Epochs	800	1200	1000
Backbone	ResNet18	ResNet50	ResNet18
Embedding Dim	512	2048	512
Projection Dim	2048	128	128

Table 10: Hyperparameters for our reproduced Viewmaker Tamkin et al. (2020) experiments.

	CIFAR10	CIFAR100	TinyImageNet
Optimizer	SGD (for encoder), Adam (for Viewmaker module)		
Learning Rate	0.015	0.06	0.06
Weight Decay	1e-4	1e-4	1e-4
Momentum	0.9	0.9	0.9
Cosine Decay	✗	✗	✗
Batch Size	128	512	128
SimCLR Loss	Viewmaker	Viewmaker	Viewmaker
Temperature	0.07	0.1	0.5
α for A2 loss	0.14	0.1	0.5
Epochs	200	800	800
Backbone	ResNet18 (small)	ResNet50	ResNet18
Noise Dim	100	100	100
Embedding Dim	512	2048	512
Projection Dim	128	128	128

B Details of StyleGAN and In-Domain GAN Inversion

StyleGAN We use StyleGAN2[**] for our experiments on CIFAR and TinyImageNet. The StyleGAN generator consists of two key components: (1) a mapping function $g_1 : \mathcal{S} \mapsto \mathcal{W}$ that maps the Gaussian-distributed latent code $s \in \mathcal{S}$ into a collection of style codes $w \in \mathcal{W}$, and (2) a generator $g : \mathcal{W} \mapsto \mathcal{X}$ that decodes $w \in \mathcal{W}$ to an image. Here, w is a concatenation of $w_1, ..., w_k$, where each w_i corresponds to the style code from the i^{th} convolutional block of g.

In-Domain GAN Inversion Let $e : \mathcal{X} \mapsto \mathcal{W}$ denote an inverter neural network. Let $d : \mathcal{X} \mapsto \mathbb{R}$ denote the discriminator network. In-domain GAN inversion Zhu et al. (2020) aims to learn a mapping from images to latent space. The encoder is trained to reconstruct real images (thus are "in-domain") and guided by image-level loss terms, *i.e.*, pixel MSE, VGG perceptual loss, and discriminator loss:

$$L_{\text{idinv}}(e, d, g) = \mathbb{E}_{x \sim P_X}\left[\|x - g \circ e(x)\|_2 + \lambda_{\text{vgg}}\|h(x) - h \circ g \circ e(x)\|_2 - \lambda_{\text{adv}} a(-\tilde{d} \circ g \circ e(x))\right], \tag{5}$$

where h is perception network and here we keep the same as in-domain inversion as VGG network, $\mathcal{A}$ is the *activation function* and $\tilde{d}$ is the *logit* or discriminator's output before activation. Note that choosing $a(t) = \text{softplus}(t) = \log\left(1 + \exp\left(t\right)\right)$ recovers the original GAN formulation Goodfellow et al. (2014a); Karras et al. (2019), and the resulting objective minimizes the Jensen-Shannon divergence between real and generated data distributions. After encoder training, we optimize the

[**] https://github.com/rosinality/stylegan2-pytorch

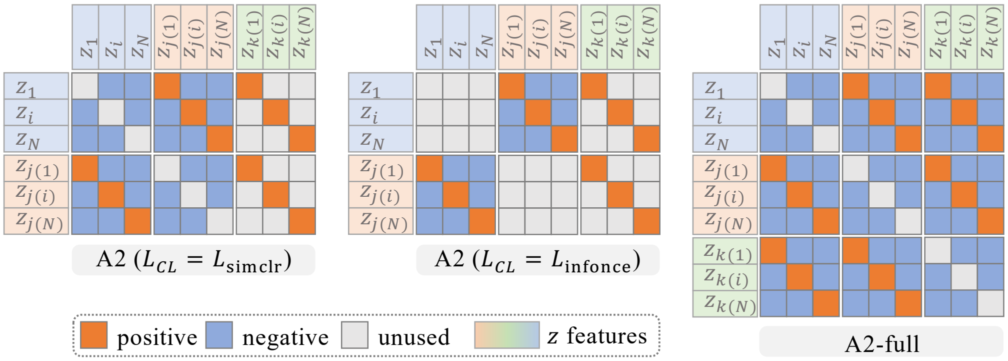

z_1 z_i z_N $z_{j(1)}$ $z_{j(i)}$ $z_{j(N)}$ $z_{k(1)}$ $z_{k(i)}$ $z_{k(N)}$
A2 ($L_{CL} = L_{\text{simclr}}$)
A2 ($L_{CL} = L_{\text{infonce}}$)
A2-full
positive
negative
unused
z features

Figure 5: Visual illustration of the three contrastive loss functions: A2-SimCLR, A2-InfoNCE, and A2-full for a mini-batch size of N images (visualizing $N = 3$). For their math formulations, please refer to Eq. (15), the Supplementary, and Eq. (13), respectively. Each small square represents the inner-product between corresponding features. The contrastive loss for each instance i is defined on the corresponding row of inner-products.

associated latent variable w for each image x with the same loss function using $w = e(x)$ as a warm start. Note that in the main text we reload the notation $e(\cdot)$ as the final results after w-optimization, which are precomputed and cached.

C PROOF OF THEOREM 3.3

Proposition C.1. *Let $\mathcal{W} = \mathbb{S}^{K-1}$, $p(z) = |\mathcal{W}|^{-1}$, and conditional $p(\tilde{z}|z)$ is a von Mises-Fisher (vMF) distribution:*

$$p(\tilde{z}|z) = C_p^{-1} e^{\tilde{z}^\top z/\tau}, \quad \textit{with} \quad C_p := \int e^{\tilde{z}^\top z/\tau} d\tilde{z}. \tag{6}$$

Then the mutual information $I(\tilde{Z}; Z) = \frac{1}{\tau}\mathbb{E}_{z,\tilde{z}}\left[z^\top \tilde{z}\right] + const.$

Proof.

$$\begin{aligned} I(z;\tilde{z}) &= KL(p(z,\tilde{z})||p(z)p(\tilde{z})) \qquad (7)\\ &= \mathbb{E}_{z\sim p(z)} KL(p(\tilde{z}|z)||p(z)) \qquad (8)\\ &= \mathbb{E}_{z\sim p(z)} \mathbb{E}_{\tilde{z}\sim p(\tilde{z}|z)} \log \frac{p(\tilde{z}|z)}{p(z)} \qquad (9)\\ &= \mathbb{E}_{z\sim p(z)} \mathbb{E}_{\tilde{z}\sim p(\tilde{z}|z)} \frac{1}{\tau} z^\top \tilde{z} - \log C_p + \log|\mathcal{Z}| \qquad (10)\\ &= \frac{1}{\tau} \mathbb{E}_{z,\tilde{z}\sim p(\tilde{z},z)} z^\top \tilde{z} + const \qquad (11) \end{aligned}$$

□

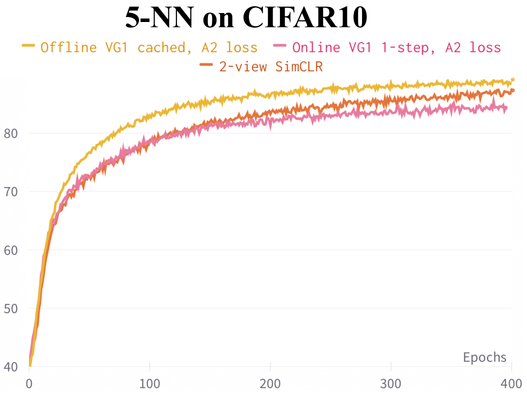


Figure 6: Online $\mathcal{W}$-*search*. In legend, VG1 refers to $\mathcal{W}$-*search*.

D Details of Training Loss

We provide an illustrative visualization of our A2 losses in Figure 5. The detailed formulation of loss functions are as follows,

$$L_{\text{infonce}} = -\sum_{i\in\mathcal{I}_1}\log\frac{\exp(z_i\cdot z_{j(i)}/\tau)}{\sum_{a\in\mathcal{I}_2}\exp(z_i\cdot z_a/\tau)} - \sum_{i\in\mathcal{I}_2}\log\frac{\exp(z_i\cdot z_{j(i)}/\tau)}{\sum_{a\in\mathcal{I}_1}\exp(z_i\cdot z_a/\tau)} \tag{12}$$

$$L_{\text{A2-full}} = -\sum_{i\in\mathcal{I}}\frac{-1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp(z_i\cdot z_p/\tau)}{\sum_{a\in\mathcal{A}(i)}\exp(z_i\cdot z_a/\tau)}, \quad \text{where } P(i)=\{j(i)\}\cup\{k(i)\} \tag{13}$$

$$L_{\text{A2-simclr}} = L_{\text{simclr}} - \sum_{i\in\mathcal{I}}\frac{\alpha}{|k(i)|}\sum_{p\in k(i)} z_i\cdot z_p/\tau \tag{14}$$

$$= -\sum_{i\in\mathcal{I}}\log\frac{\exp(z_i\cdot z_{j(i)}/\tau + \frac{\alpha}{|k(i)|}\sum_{p\in k(i)}(z_i\cdot z_p)/\tau)}{\sum_{a\in\mathcal{A}(i)}\exp(z_i\cdot z_a/\tau)} \tag{15}$$

In L_{infonce}, $\mathcal{I}_1$ and $\mathcal{I}_2$ are the set of indices of two positive views. An ablation of A2 losses is provided in Table 11. We observe that A2-InfoNCE performs the best for both datasets. We used A2-InfoNCE as our A2 loss if not specified.

Table 11: Ablation on the loss functions with VG1 views on the CIFAR10 and CIFAR100 datasets. A2-SimCLR is $L_{\text{simclr}} - L_{\text{align}}$ and A2-InfoNCE is $L_{\text{infonce}} - L_{\text{align}}$.

	CIFAR10		CIFAR100	
Loss	Acc@1	5-NN	Acc@1	5-NN
A2-full	92.57	91.57	71.82	66.06
A2-SimCLR ($\alpha = 0.5$)	92.66	91.05	72.27	65.85
A2-InfoNCE ($\alpha = 0.5$)	**92.90**	90.95	**72.76**	66.46
A2-InfoNCE ($\alpha = 1$)	92.54	90.80	72.61	66.96

E Online $\mathcal{W}$-search

In the online $\mathcal{Z}$-search setting, the optimization is performed involving the current SimCLR encoder during training. We tried to perform 1-step optimization with fast sign gradient Goodfellow et al. (2014b), but observed the results are worse than the SimCLR baseline. The 5-NN accuracies are reported in Figure 6.

Table 12: Linear-probing and 5-NN accuracies of SimSiam experiments on CIFAR10.

	Acc@1	5-NN
SimSiam baseline	90.17	89.27
$\mathcal{W}$-*search*-SimCLR + A2-SimSiam	**90.96**	89.81
$\mathcal{W}$-*search*-SimSiam + A2-SimSiam	90.79	89.82
$\mathcal{W}$-*perturb* + A2-SimSiam	90.28	**89.99**
SimCLR baseline	92.04	90.65
$\mathcal{W}$-*search*-SimSiam + A2-InfoNCE	**92.44**	90.88

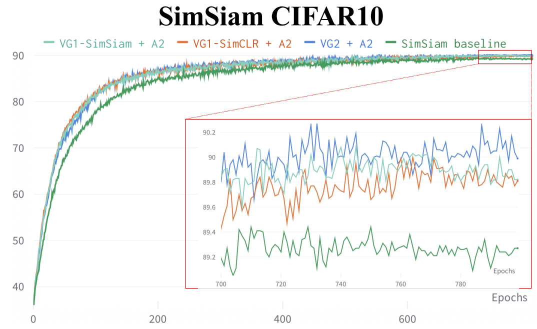


Figure 7: 5-NN accuracy curves of SimSiam experiments on CIFAR10. In legend, VG1 refers to $\mathcal{W}$-*search* and VG2 refers to $\mathcal{W}$-*perturb*.

F SimSiam Experiments

We conducted experiments of SimSiam Chen & He (2021) with generated views. The 5-NN accuracy curves are reported in Fig. 7. The linear probe accuracy is reported in Table 12. The A2 loss needs to be adjusted accordingly to avoid training collapse,

$$L_{\text{A2-simsiam}} = L_{\text{simsiam}} + \sum_{i \in I} \frac{\alpha}{|k(i)|} \sum_{p \in k(i)} D(\texttt{predictor}(z_p), \texttt{stopgrad}(z_i)) \tag{16}$$

where D is the cosine similarity and L_{simsiam} is the SimSiam loss function.

G Ablation on Hyperparameters

In this section we provide ablations on hyperparameters ϵ_1, ϵ_2, and λ introduced in Eq. (3). In addition, we perform grid search on σ introduced in Sec. 3.2.

Ablation on ϵ_1 and ϵ_2. We conduct ablation studies of ϵ_1, ϵ_2, and λ on CIFAR10. In Table 13, we set $\epsilon_2 = \epsilon_1 + 0.2$ except for ϵ_1 of values 0.1 and 0.2. We empirically find that it is difficult to reach a large pairwise distance ϵ_2 when ϵ_1 is small, and a large ϵ_2 leads to more optimization steps. By design, a large ϵ_2 encourages generating diverse samples. A *rule of thumb* is to set $\epsilon_2 \geq \epsilon_1$.

Table 13: Ablation on ϵ_1 and ϵ_2, $\lambda = 0.01$. Experiments are conducted on CIFAR10.

ϵ_1	ϵ_2	Acc@1
0.1	0.15	92.584 ± 0.023
0.2	0.35	92.615 ± 0.048
0.3	0.50	**92.898** ± 0.045
0.5	0.70	92.451 ± 0.053
0.7	0.90	91.760 ± 0.032
0.9	1.10	91.561 ± 0.051

Table 14: Ablation on ϵ_1, $\epsilon_2 = 0.15$ and $\lambda = 0.01$. Experiments are conducted on CIFAR10 with the A2 loss.

ϵ_1	ϵ_2	Acc@1
0.1	0.15	92.584 ± 0.023
0.2	0.15	92.385 ± 0.043
0.3	0.15	**92.643** ± 0.043
0.5	0.15	92.455 ± 0.032
0.7	0.15	92.174 ± 0.029

Table 15: Ablation studies on λ. For all entries we fix $\epsilon_1 = 0.3$ and $\epsilon_2 = 0.5$. Experiments are conducted on CIFAR10 with the A2 loss.

ϵ_1	λ	Acc@1
0.3	0	92.555 ± 0.028
0.3	0.005	92.756 ± 0.042
0.3	0.01	**92.898** ± 0.045
0.3	0.02	92.854 ± 0.027

Table 16: Ablation on σ of value 0, 0.1, 0.2, 0.4 and 1.0. Experiments are conducted on CIFAR10 with A2 loss.

View	Acc@1
$e(x)$	91.697 ± 0.038
$e(x) + w_{\texttt{Gauss}} \sim N(0, 0.1)$	92.297 ± 0.024
$e(x) + w_{\texttt{Gauss}} \sim N(0, 0.2)$	92.383 ± 0.033
$e(x) + w_{\texttt{Gauss}} \sim N(0, 0.4)$	91.852 ± 0.030
$e(x) + w_{\texttt{Gauss}} \sim N(0, 1.0)$	87.893 ± 0.026
$e(x) + \texttt{proj}(w_{\texttt{Gauss}}) \sim N(0, 0.2)$	**92.479** ± 0.055

Table 17: Ablation on σ of value 0.1, 0.2, 0.4 and 1.0. Experiments are conducted on CIFAR100 with A2 loss.

View	Acc@1	5-NN
$e(x) + w_{\texttt{Gauss}} \sim N(0, 0.1)$	71.84	66.33
$e(x) + w_{\texttt{Gauss}} \sim N(0, 0.2)$	**72.95**	**66.60**
$e(x) + w_{\texttt{Gauss}} \sim N(0, 0.4)$	71.69	65.33
$e(x) + w_{\texttt{Gauss}} \sim N(0, 1.0)$	68.05	61.06

Ablation on λ. In Table 14, we fix $\epsilon_2 = 0.15$ and vary ϵ_1. In Table 15, we fix $\epsilon_1 = 0.3$ and $\epsilon_2 = 0.5$ and vary λ.

Ablation on σ. We perform grid search on σ for $\mathcal{W}$*-perturb* and report results in Tables 16 and 17. We find that for both CIFAR10 and CIFAR100, $\sigma = 0.2$ leads to the best results, which is consistent with the empirical findings in Jahanian et al. (2021).

H Ablation on Appended Views

We perform ablations on the appended views (with A2 loss) on CIFAR10. As shown in Table 18, appending the in-domain GAN inversion reconstructed images, optimizing Eq. (3) with SGD (instead of Adam), optimizing Equation 2 for only 1 or 2 steps, and appending the expert or the original view lead to inferior performance than $\mathcal{W}$*-search*.

I More Results on Replacing Expert Views

Augmented GAN training. Please note that for all experiments in Section 5.3 on replacing expert transformations, we employ *augmented* GAN training, *i.e.*, the StyleGAN2 generator is trained with expert augmentations. Please also note that this augmentation is different from the *differentiable augmentations* commonly used in training GAN with limited data Karras et al. (2020a); Zhao et al. (2020). Here we apply expert transforms on real images with probability 0.5, thus the expert transformations will be *leaked* to the generator.

Additional results. For the following additional results in Table 19, we use `-aug` to denote augmented GAN training[††]. As shown in rows 2 and 3, 7 and 8, in Table 19, the A2 loss is better than A2-full. Note that row 3 is equivalent to online 3-view SimCLR with *basic* transforms. From row 1-4 we found that $\mathcal{W}$*-search*-aug views significantly improves basic transforms.

J Additional Samples of Generated Views

Here we show additional visual samples via $\mathcal{W}$*-search* and $\mathcal{W}$*-perturb* in Figure 8, 9, and 10. We see that for CIFAR10, CIFAR100 and TinyImageNet, $\mathcal{W}$*-search* tends to generate more diverse samples than $\mathcal{W}$*-perturb*.

[††] To ease the notation, we remove `-aug` for all entries of Tab. 5.

Table 18: Ablation on *appended* views. Experiments are conducted on CIFAR10 with A2 loss.

View	Acc@1
Reconstruction $g \circ e(x)$	91.697 ± 0.038
SGD Optimizer	91.674 ± 0.045
1-Step Optimization	92.467 ± 0.029
2-Step Optimization	92.359 ± 0.018
Original	91.936 ± 0.017

Table 19: Ablation on different configurations on complete replacement of expert views in the CIFAR100 dataset. $\mathcal{W}$-*search*-aug indicates that the generator is trained with expert *augmentations*.

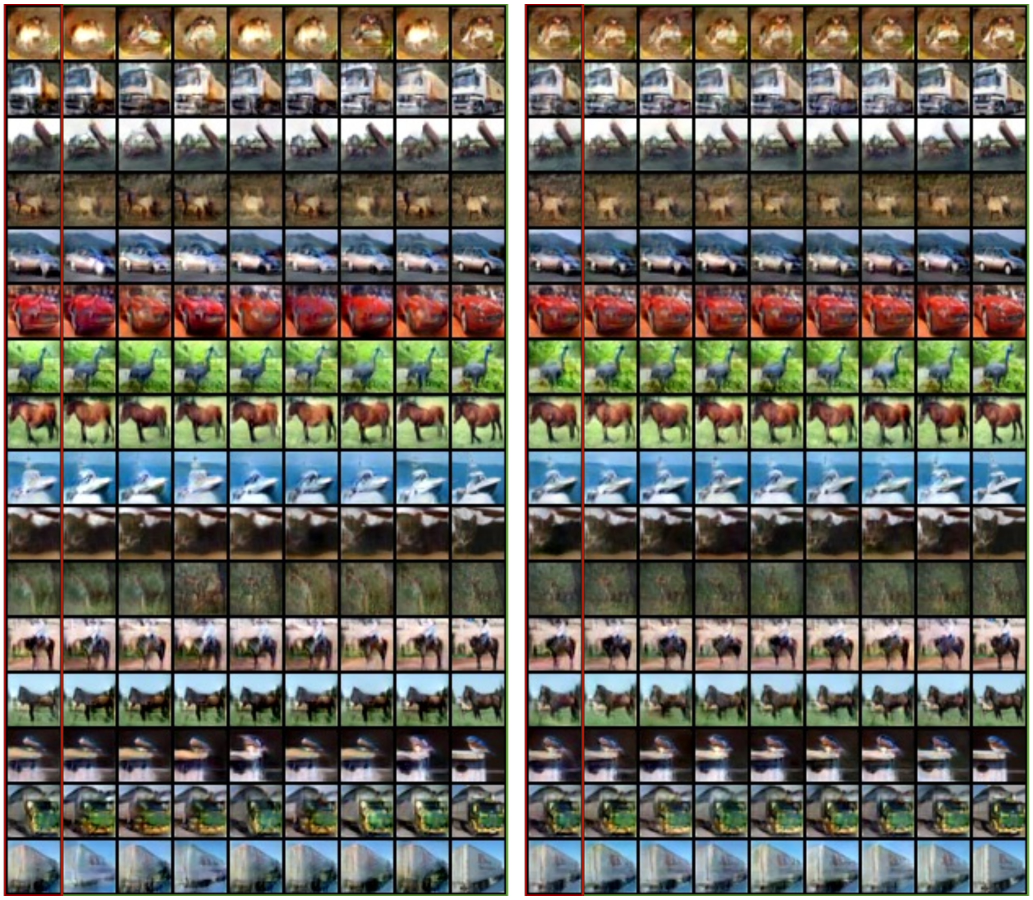
Row	View 1	View 2	View 3	Loss	Acc@1	5-NN
1	basic	basic	✗	SimCLR	47.61	35.86
2	basic	basic	basic	A2	49.35	38.03
3	basic	basic	basic	A2-full	44.64	34.18
4	basic	basic	$\mathcal{W}$-*search*-aug	A2	**69.55**	**63.65**
5	basic	basic	$\mathcal{W}$-*search*-aug	A2-full	69.16	63.15
6	basic	$\mathcal{W}$-*search*-aug	✗	SimCLR	69.19	62.36
7	$\mathcal{W}$-*search*-aug	$\mathcal{W}$-*search*-aug	$\mathcal{W}$-*search*-aug	A2	62.80	53.26
8	$\mathcal{W}$-*search*-aug	$\mathcal{W}$-*search*-aug	$\mathcal{W}$-*search*-aug	A2-full	61.17	52.09
9	$\mathcal{W}$-*search*-aug	$\mathcal{W}$-*search*-aug	✗	SimCLR	57.65	49.27
10	$\mathcal{W}$-*search*	$\mathcal{W}$-*search*	✗	SimCLR	47.50	37.54

K Additional Training Curves

In Figure 11 and 12, we provide additional 5-NN accuracies evaluated after every epoch during training.

L Ethical Impact

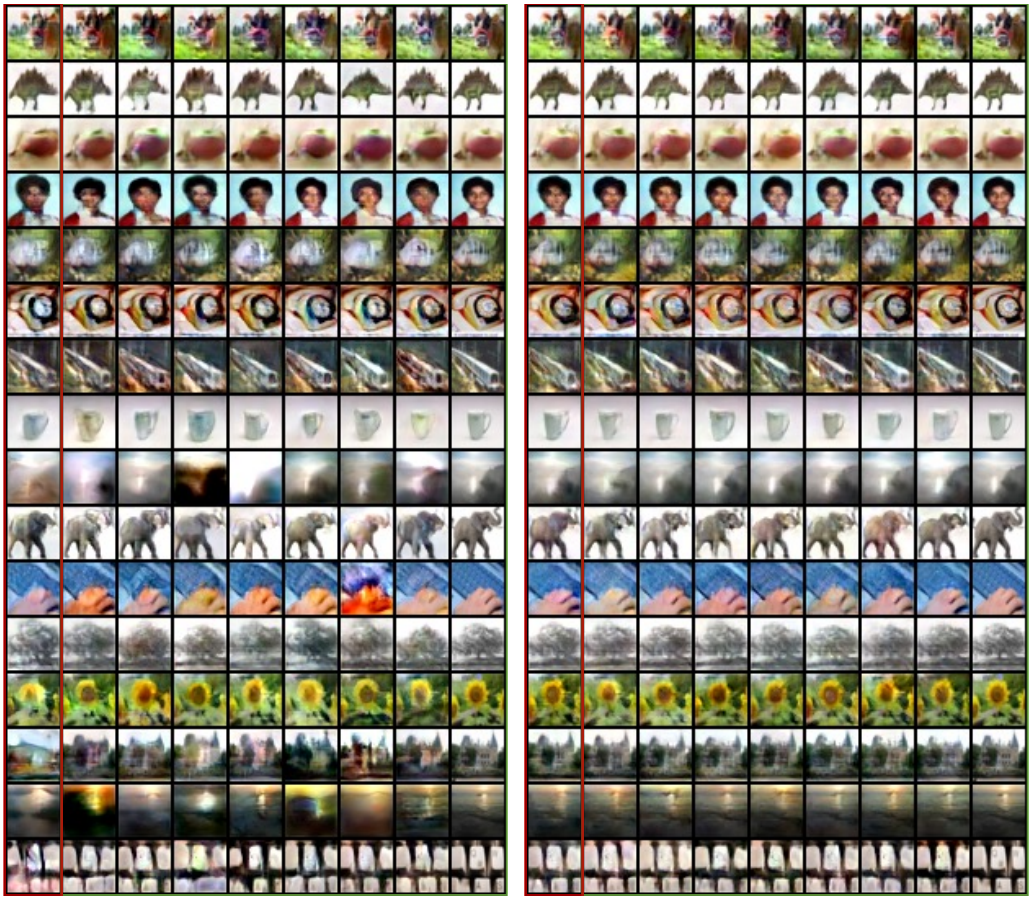
In this work, we utilized generative models for learning augmentations of natural images, which potentially relate to image generation techniques. As with any good generative model of image data, there is risk that work built on these generative models could potentially be used for the creation of deliberately deceptive imagery. However, our work focuses on an orthogonal direction of representation learning and conducts an extensive empirical study on how to generate meaningful views and how to assimilate them.



(a) $\mathcal{W}$-*search*

(b) $\mathcal{W}$-*perturb*

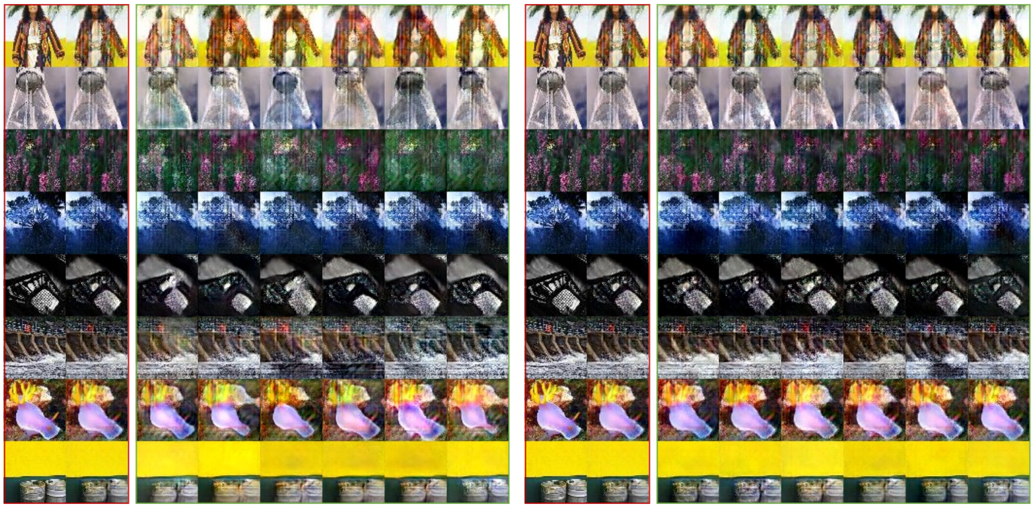
Figure 8: Visual samples of $\mathcal{W}$-*search* and $\mathcal{W}$-*perturb* on CIFAR10 dataset. In each subfigure panel, the first column (in the red box) is the original image, and columns 2-9 are generated 8 views.



(a) $\mathcal{W}$-*search*

(b) $\mathcal{W}$-*perturb*

Figure 9: Visual samples of $\mathcal{W}$-*search* and $\mathcal{W}$-*perturb* on CIFAR100 dataset. In each subfigure panel, the first column (in the red box) is the original image, and columns 2-9 are generated 8 views.



(a) $\mathcal{W}$-*search*

(b) $\mathcal{W}$-*perturb*

Figure 10: Visual samples of $\mathcal{W}$-*search* and $\mathcal{W}$-*perturb* on TinyImageNet dataset. In each subfigure panel, the first two columns are the original image and its reconstruction, and columns 3-8 are generated 6 views.

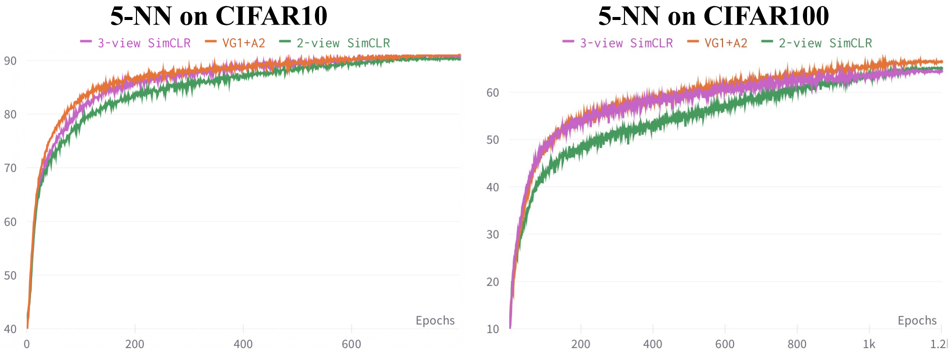
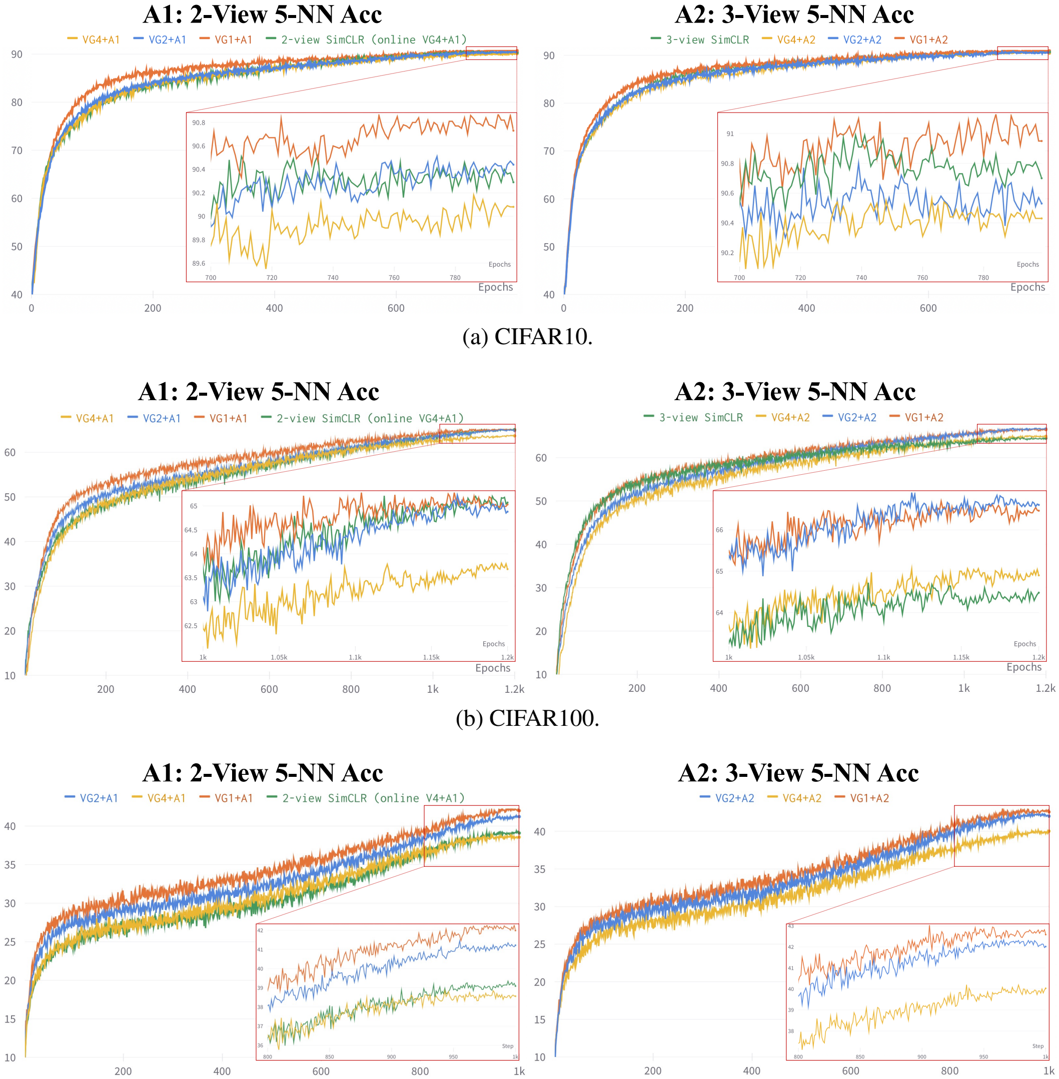


Figure 11: 5-NN accuracy curves during training. In legend, VG1 refers to $\mathcal{W}$-*search* and VG2 refers to $\mathcal{W}$-*perturb*.



(a) CIFAR10.

(b) CIFAR100.

(c) TinyImageNet.

Figure 12: 5-NN accuracy curves during training. In legend, VG1 refers to $\mathcal{W}$*-search* and VG2 refers to $\mathcal{W}$*-perturb*.