Unsupervised Learning of Structured Representations via Closed-Loop Transcription

Shengbang Tong[1,*], Xili Dai[1,2,†], Yubei Chen[3], Mingyang Li[5], Zengyi Li[1], Brent Yi[1], Yann LeCun[3,4], Yi Ma[1,5]
[1]University of California, Berkeley, [2]Hong Kong University of Science and Technology (Guangzhou), [3]Center for Data Science, New York University, [4]Courant Institute, New York University, [5]Tsinghua-Berkeley Shenzhen Institute (TBSI)

This paper proposes an unsupervised method for learning a unified representation that serves both discriminative and generative purposes. While most existing unsupervised learning approaches focus on a representation for only one of these two goals, we show that a unified representation can enjoy the mutual benefits of having both. Such a representation is attainable by generalizing the recently proposed *closed-loop transcription* framework, known as CTRL, to the unsupervised setting. This entails solving a constrained maximin game over a rate reduction objective that expands features of all samples while compressing features of augmentations of each sample. Through this process, we see discriminative low-dimensional structures emerge in the resulting representations. Under comparable experimental conditions and network complexities, we demonstrate that these structured representations enable classification performance close to state-of-the-art unsupervised discriminative representations, and conditionally generated image quality significantly higher than that of state-of-the-art unsupervised generative models.

1. Introduction

In the past decade, we have witnessed an explosive development in the practice of machine learning, particularly with deep learning methods. A key driver of success in practical applications has been marvelous engineering endeavors, often focused on fitting increasingly large deep networks to input data paired with task-specific sets of labels. Brute-force approaches of this nature, however, exert tremendous demands on hand-labeled data for supervision and computational resources for training and inference. As a result, an increasing amount of attention has been directed toward using self-supervised or unsupervised techniques to learn representations that can not only learn without human annotation effort, but also be shared across downstream tasks.

Discriminative versus Generative. Tasks in unsupervised learning are typically separated into two categories. Discriminative ones frame high-dimensional observations as inputs, from which low-dimensional class or latent information can be extracted, while generative ones frame observations as generated outputs, which should often be sampled given some semantically meaningful conditioning.

Unsupervised learning approaches targeted at discriminative tasks are mainly based on a key idea: to pull different views from the same instance closer while enforcing a non-collapsed representation by either contrastive learning techniques [1–3], covariance regularization methods [4, 5], or using architecture design [6, 7]. Their success is typically measured by the accuracy of a simple classifier (say a shallow network) trained on the representations that they produce, which have progressively improved over the years. Representations learned from these approaches, however, do not emphasize much about the intrinsic structure of the data distribution, and have not demonstrated success for generative purposes.

In parallel, generative methods like GANs [8], VAEs [9] and diffusion models [10] have also been explored for unsupervised learning. Although generative methods have made striking progress in the quality of the sampled or autoencoded data, when compared to the aforementioned discriminative methods, representations learned with these approaches demonstrate inferior performance in classification.

[*]Equal contribution
[†]Work done during visiting at Berkeley

First Conference on Parsimony and Learning (CPAL 2024).

Toward A Unified Representation? The disparity between discriminative and generative approaches in unsupervised learning, contrasted against the fundamental goal of learning representations that are useful across many tasks, leads to a natural question that we investigate in this paper: *in the unsupervised setting, is it possible to learn a unified representation that is effective for both discriminative and generative purposes? Further, do they mutually benefit each other?* Concretely, we aim to learn a *structured representation* with the following two properties:

1. The learned representation should be discriminative, such that simple classifiers applied to learned features yield high classification accuracy.
2. The learned representation should be generative, with enough diversity to recover raw inputs, and structure that can be exploited for sampling and generating new images.

The fact that human visual memory serves both discriminative tasks (for example, detection and recognition) and generative or predictive tasks (for example, via replay) [11–13] indicates that this goal is achievable. Beyond being possible, these properties are also highly practical – successfully completing generative tasks like unsupervised conditional image generation [14], for example, inherently requires that learned features for different classes be both structured for sampling and discriminative for conditioning. On the other hand, the generative property can serve as a natural regularization to avoid representation collapse.

Closed-Loop Transcription via a Constrained Maximin Game. The class of linear discriminative representations (LDRs) has recently been proposed for learning diverse and discriminative features for multi-class (visual) data, via optimization of the *rate reduction* objective [15]. In the supervised setting, these representations have been shown to be be both discriminative and generative if learned in *a closed-loop transcription* framework via a maximin game over the rate reduction utility between an encoder and a decoder [16]. Beyond the standard joint learning setting, where all classes are sampled uniformly throughout training, the closed-loop framework has also been successfully adapted to the *incremental setting* [17], where the optimal multi-class LDR is learned one class at a time. In the incremental (supervised) learning setting, one solves a *constrained maximin problem* over the rate reduction utility which keeps learned memory of old tasks intact (as constraints) while learning new tasks. It has been shown that this new framework can effectively alleviate the catastrophic forgetting suffered by most supervised learning methods.

Contributions. In this work, we show that the closed-loop transcription framework proposed for learning LDRs in the supervised setting [15] can be adapted to a purely unsupervised setting. In the unsupervised setting, we only have to view each sample and its augmentations as a "new class" while using the rate reduction objective to ensure that learned features are both *invariant* to augmentation and *self-consistent* in generation; this leads to a constrained maximin game that is similar to the one explored for incremental learning [17]. Our overall approach is illustrated in Figure 1.

As we experimentally demonstrate in Section 4, our formulation benefits from the mutual benefits of both discriminative and generative properties. It bridges the gap between two formerly distinct set of methods: by standard metrics and under comparable experimental conditions, it enables classification performance on par with and unsupervised conditional generative quality significantly higher than state-of-the-art techniques. Coupled with evidence from prior work, this suggests that the closed-loop transcription through the (constrained) maximin game between the encoder and decoder has the potential to offer a unifying framework for both discriminative and generative representation learning, across supervised, incremental, and unsupervised settings.

Method	Linear Probe	Image Generation	UCIG
SimCLR [1]	✔	✗	✗
MOCO-V2 [2]	✔	✗	✗
ContraD [18]	✔	✔	✗
PATCH-VAE [19]	✔	✔	✗
CTRL-Binary [16]	✔	✔	✗
SLOGAN [14]	✗	✔	✔
U-CTRL (ours)	✔	✔	✔

Table 1: Comparison of the downstream task capabilities of different unsupervised learning methods. UCIG refers to Unsupervised Conditional Image Generation [14].

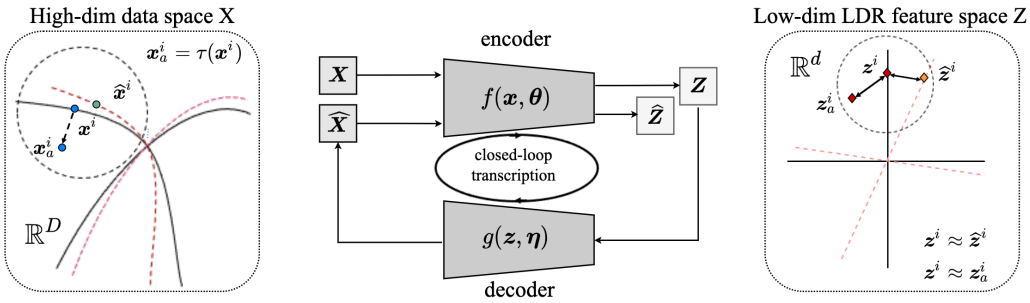


Figure 1: **Overall framework** of closed-loop transcription for unsupervised learning. Two additional constraints are imposed on the Binary-CTRL method proposed in prior work [16]: 1) self-consistency for sample-wise features $\boldsymbol{z}^i$ and $\hat{\boldsymbol{z}}^i$, say $\boldsymbol{z}^i \approx \hat{\boldsymbol{z}}^i$; and 2) invariance/similarity among features of augmented samples $\boldsymbol{z}^i$ and $\boldsymbol{z}_a^i$, say $\boldsymbol{z}^i \approx \boldsymbol{z}_a^i = f(\tau(\boldsymbol{x}^i), \theta)$, where $\boldsymbol{x}_a^i = \tau(\boldsymbol{x}^i)$ is an augmentation of sample $\boldsymbol{x}^i$ via some transformation $\tau(\cdot)$.

2. Related Work

Our work is mostly related to three categories of unsupervised learning methods: (1) self-supervised learning via discriminative models, (2) self-supervised learning via generative models, and (3) unsupervised conditional image generation. Table 1 compares the capabilities of models learned by various representative unsupervised learning methods.

Self-Supervised Learning for Discriminative Models. On the discriminative side, works like SimCLR [1], MoCo [2], and BYOL [3] have recently shown overwhelming effectiveness in learning discriminative representations of data. MoCo [2] and SimCLR [1] seek to learn features by pulling together features of augmented versions of the same sample while pushing apart features of all other samples, while BYOL [3] trains a student network to predict the representation of a teacher network in a contrastive setting. BarlowTwins [5] and TCR [20] learn by regularizing the covariance matrix of the embedding. However, features learned by this class of methods are typically highly compressed, and *not* designed to be used for generative purposes.

Self-Supervised Learning with Generative Models. On the generative side, the original GAN [8] can be viewed as a natural self-supervised learning task. With an additional linear probe, works like DCGAN [21] have shown that features in the discriminator can be used for discriminative tasks. To further enhance the features, extensions like BiGAN [22] and ALI [23] introduce a third network into the GAN framework, aimed at learning an inverse mapping for the generator, which when coupled with labeled images can be used to study and supervise semantics in learned representations. Other works like SSGAN [24], SSGAN-LA [25], and ContraD [18] propose to put augmentation tasks into GAN training to facilitate representation learning. Outside of GANs, variational autoencoders (VAEs) have been adapted to generate more semantically meaningful representations by trading off latent channel capacity and independence constraints with reconstruction accuracy [26], an idea that has also been incorporated into recognition improvements using patch-level bottlenecks [27], which encourage a VAE to focus on useful patterns in images. By incorporating data-augmentation, VAE is also shown to achieve fair discriminative performance [28]. Recently, works like MAE [29] and CAE [30] have learned representations by solving masked reconstruction tasks using vision transformers. Autogressive approaches like iGPT [31] have also demonstrated decent self-supervised learning performance, which improves further with the incorporation of contrastive learning [32]. However, unless supervised, features learned by those previously mentioned methods either do not have strong discriminative performance, or cannot be directly exploited to condition the generative task.

Unsupervised Conditional Image Generation (UCIG). For generative models, we often want to be able to generate images conditioned on a certain class or style, even in a completely unsupervised setting. This requires that the learned representations have structures that correspond to the desired conditioning. InfoGAN [33] proposes to learn interpretable representations by maximizing the mutual information between the observation and a subset of the latent code. ClusterGAN [34] assumes a discrete Gaussian prior where discrete variables are defined as a one-hot vector and continuous variables are sampled from Gaussian distribution. Self-Conditioned GAN [35] uses clustering of discriminative features as labels to train. SLOGAN [14] proposes a new conditional contrastive loss (U2C) to learn latent distribution of the data. Note that compared

to our work, ClusterGAN and SLOGAN introduce an additional encoder that leads to increased computational complexity. On the VAE side, works like VaDE [36] cluster based on the learned feature of a supervised ResNet. Variational Cluster [37] simultaneously learns a prior that captures the latent distribution of the images and a posterior to help discriminate between data points in an end-to-end unsupervised setting. In this work, we will see how clusters can be estimated in a principled way in a more unified framework, by optimizing the same type of objective function that we use for learning features.

3. Method

3.1. Preliminaries: Rate Reduction and Closed-Loop Transcription

Assumptions on Data. Our work, as well as prior work in closed-loop transcription [16, 17], considers a set of N images $\boldsymbol{X} = [\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^N] \subset \mathbb{R}^D$ sampled from k classes. Borrowing notation from [38], the membership of the N samples in the k classes is denoted using k diagonal matrices: $\boldsymbol{\Pi} = \{\boldsymbol{\Pi}_j \in \mathbb{R}^{N\times N}\}_{j=1}^k$, where the diagonal entry $\boldsymbol{\Pi}_j(i,i)$ of $\boldsymbol{\Pi}_j$ is the probability of sample i belonging to subset j. Let $\Omega \doteq \{\boldsymbol{\Pi} \mid \sum \boldsymbol{\Pi}_j = \boldsymbol{I}, \boldsymbol{\Pi}_j \geq \boldsymbol{0}.\}$ be the set of all such matrices. WLOG, we may assume that classes are separable, with images for each belonging to a low-dimensional submanifold in the space $\mathbb{R}^D$.

Unsupervised Discriminative Autoencoding. The goal of transcription is to learn a unified representation, with the structure required to both classify and generate images from these k classes. Concretely, this is achieved by learning two continuous mappings: (1) an encoder parametrized by θ: $f(\cdot,\theta) : \boldsymbol{x} \mapsto \boldsymbol{z} \in \mathbb{R}^d$ with $d \ll D$ such that all samples are mapped to their features as $\boldsymbol{X} \xrightarrow{f(\boldsymbol{x},\theta)} \boldsymbol{Z}$ with $\boldsymbol{Z} = [\boldsymbol{z}^1, \boldsymbol{z}^2, ..., \boldsymbol{z}^N] \subset \mathbb{R}^d$, and (2) an inverse map $g(\cdot,\eta) : \boldsymbol{z} \mapsto \hat{\boldsymbol{x}} \in \mathbb{R}^D$ such that $\boldsymbol{x}$ and $\hat{\boldsymbol{x}} = g(f(\boldsymbol{x}))$ is close. In other words, $\boldsymbol{X} \xrightarrow{f(\boldsymbol{x},\theta)} \boldsymbol{Z} \xrightarrow{g(\boldsymbol{z},\eta)} \hat{\boldsymbol{X}}$ forms an autoencoding.

In this work, we specifically learn this mapping in an *entirely unsupervised* fashion, without knowing the ground-truth class labels $\boldsymbol{\Pi}$ at all. As stated in the introduction, a both discriminative and generative representation is difficult to achieve by standard generative methods like VAEs and GANs. This is one of the motivations for the closed-loop transcription framework (CTRL) proposed by [16], which we will generalize to the unsupervised setting.

Maximizing Rate Reduction. The CTRL framework [16] was proposed for the supervised setting, where it aims to map each class onto an independent linear subspace. As shown in [38], such a linear discriminative representation (LDR) can be achieved by maximizing a coding rate reduction objective, known as *the MCR*^{2} *principle*:

$$\Delta R(\boldsymbol{Z}|\boldsymbol{\Pi}) \doteq \underbrace{\frac{1}{2}\log\det\left(\boldsymbol{I} + \frac{d}{N\epsilon^2}\boldsymbol{Z}\boldsymbol{Z}^\top\right)}_{R(\boldsymbol{Z})} - \underbrace{\sum_{j=1}^{k}\frac{\mathsf{tr}(\boldsymbol{\Pi}_j)}{2N}\log\det\left(\boldsymbol{I} + \frac{d}{\mathsf{tr}(\boldsymbol{\Pi}_j)\epsilon^2}\boldsymbol{Z}\boldsymbol{\Pi}_j\boldsymbol{Z}^\top\right)}_{R^c}. \tag{1}$$

where each $\boldsymbol{\Pi}_j$ encodes the membership of the N samples described before. As discussed in [15], the first term $R(\boldsymbol{Z})$ measures the total rate (volume) of all features whereas the second term R^c measures the average rate (volume) of the k components. Our work adapts this formula to design meaningful objectives in the unsupervised setting.

Closed-Loop Transcription. To learn the autoencoding $\boldsymbol{X} \xrightarrow{f(\boldsymbol{x},\theta)} \boldsymbol{Z} \xrightarrow{g(\boldsymbol{z},\eta)} \hat{\boldsymbol{X}}$, a fundamental question is how we measure the difference between $\boldsymbol{X}$ and the regenerated $\hat{\boldsymbol{X}} = g(f(\boldsymbol{X}))$. It is typically very difficult to put a proper distance measure in the image space [39]. To bypass this difficulty, the *closed-loop transcription* framework [16] proposes to measure the difference between $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ through the difference between their features $\boldsymbol{Z}$ and $\hat{\boldsymbol{Z}}$ mapped through the same encoder:

$$\boldsymbol{X} \xrightarrow{f(\boldsymbol{x},\theta)} \boldsymbol{Z} \xrightarrow{g(\boldsymbol{z},\eta)} \hat{\boldsymbol{X}} \xrightarrow{f(\boldsymbol{x},\theta)} \hat{\boldsymbol{Z}}. \tag{2}$$

The difference can be measured by the rate reduction between $\boldsymbol{Z}$ and $\hat{\boldsymbol{Z}}$, a special case of (1) with $k = 2$ classes:

$$\Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}}) \doteq R(\boldsymbol{Z} \cup \hat{\boldsymbol{Z}}) - \frac{1}{2}\big(R(\boldsymbol{Z}) + R(\hat{\boldsymbol{Z}})\big). \tag{3}$$

Such a ΔR is a principled distance between subspace-like Gaussian ensembles, with the property that $\Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}}) = 0$ iff $\text{Cov}(\boldsymbol{Z}) = \text{Cov}(\hat{\boldsymbol{Z}})$ [40].

As shown in [16], applying this measure in the closed-loop CTRL formulation can already learn a decent autoencoding, even without class information. This is known as the CTRL-Binary program:

$$\max_{\theta} \min_{\eta} \quad \Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}}) \tag{4}$$

However, note that (4) is practically limited because it only aligns the dataset $\boldsymbol{X}$ and the regenerated $\hat{\boldsymbol{X}}$ at the distribution level. There is no guarantee that for each sample $\boldsymbol{x}$ would be close to the decoded $\hat{\boldsymbol{x}} = g(f(\boldsymbol{x}))$. For example, [16] shows that a car sample can be decoded into a horse; the so obtained (autoencoding) representations are not sample-wise *self-consistent*!

3.2. Sample-Wise Constraints for Unsupervised Transcription

To improve discriminative and generative properties of representations learned in the unsupervised setting, we propose two additional mechanisms for the above CTRL-Binary maximin game (4). For simplicity and uniformity, here these will be formulated as equality constraints over rate reduction measures, but in practice they can be enforced softly during optimization.

Sample-wise Self-Consistency via Closed-Loop Transcription. First, to address the issue that CTRL-Binary does not learn a sample-wise consistent autoencoding, we need to promote $\hat{\boldsymbol{x}}$ to be close to $\boldsymbol{x}$ for each sample. In the CTRL framework, this can be achieved by enforcing that their corresponding features $\boldsymbol{z} = f(\boldsymbol{x})$ and $\hat{\boldsymbol{z}} = f(\hat{\boldsymbol{x}})$ are the same or close. To promote sample-wise self-consistency, where $\hat{\boldsymbol{x}} = g(f(\boldsymbol{x}))$ is close to $\boldsymbol{x}$, we want the distance between $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$ to be zero or small, for all N samples. This can be formulated using rate reduction; note that this again avoids measuring differences in the image space:

$$\sum_{i \in N} \Delta R(\boldsymbol{z}^i, \hat{\boldsymbol{z}}^i) = 0. \tag{5}$$

Self-Supervision via Compressing Augmented Samples. Since we do not know any class label information between samples in the unsupervised settings, the best we can do is to view every sample and its augmentations (say via translation, rotation, occlusion etc) as one "class" — a basic idea behind almost all self-supervised learning methods. In the rate reduction framework, it is natural to compress the features of each sample and its augmentations. In this work, we adopt the standard transformations in SimCLR [1] and denote such a transformation as τ. We denote each augmented sample $\boldsymbol{x}_a = \tau(\boldsymbol{x})$, and its corresponding feature as $\boldsymbol{z}_a = f(\boldsymbol{x}_a, \theta)$. For discriminative purposes, we hope the classifier is *invariant* to such transformations. Hence it is natural to enforce that the features $\boldsymbol{z}_a$ of all augmentations are the same as that $\boldsymbol{z}$ of the original sample $\boldsymbol{x}$. This is equivalent to requiring the distance between $\boldsymbol{z}$ and $\boldsymbol{z}_a$, measured in terms of rate reduction again, to be zero (or small) for all N samples:

$$\sum_{i \in N} \Delta R(\boldsymbol{z}^i, \boldsymbol{z}_a^i) = 0. \tag{6}$$

3.3. Unsupervised Representation Learning via Closed-Loop Transcription

So far, we know the CTRL-Binary objective $\Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}})$ in (4) helps align the distributions while sample-wise self-consistency (5) and sample-wise augmentation (6) help align and compress features associated with each sample. Besides consistency, we also want learned representations are maximally discriminative for different samples (here viewed as different "classes"). Notice that the rate distortion term $R(\boldsymbol{Z})$ measures the coding rate (hence volume) of all features. It has been observed in [20] that by maximizing this term, learned features expand and hence become more discriminative.

Unsupervised CTRL. Putting these elements together, we propose to learn a representation via the following constrained maximin program, which we refer to as *unsupervised CTRL* (U-CTRL):

$$\begin{aligned} &\max_{\theta} \min_{\eta} \quad R(\boldsymbol{Z}) + \Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}}) \\ &\text{subject to} \quad \sum_{i \in N} \Delta R(\boldsymbol{z}^i, \hat{\boldsymbol{z}}^i) = 0, \quad \text{and} \quad \sum_{i \in N} \Delta R(\boldsymbol{z}^i, \boldsymbol{z}_a^i) = 0. \end{aligned} \tag{7}$$

In practice, the above program can be optimized by alternating maximization and minimization between the encoder $f(\cdot, \theta)$ and the decoder $g(\cdot, \eta)$. We adopt the following optimization strategy that works well in practice, which is used for all subsequent experiments on real image datasets:

$$\max_{\theta} \; R(\boldsymbol{Z}) + \Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}}) - \lambda_1 \sum_{i \in N} \Delta R(\boldsymbol{z}^i, \boldsymbol{z}_a^i) - \lambda_2 \sum_{i \in N} \Delta R(\boldsymbol{z}^i, \hat{\boldsymbol{z}}^i); \tag{8}$$

$$\min_{\eta} \; R(\boldsymbol{Z}) + \Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}}) + \lambda_1 \sum_{i \in N} \Delta R(\boldsymbol{z}^i, \boldsymbol{z}_a^i) + \lambda_2 \sum_{i \in N} \Delta R(\boldsymbol{z}^i, \hat{\boldsymbol{z}}^i), \tag{9}$$

where the constraints $\sum_{i \in N} \Delta R(\boldsymbol{z}^i, \hat{\boldsymbol{z}}^i) = 0$ and $\sum_{i \in N} \Delta R(\boldsymbol{z}^i, \boldsymbol{z}_a^i) = 0$ in (7) have been converted (and relaxed) to Lagrangian terms with corresponding coefficients λ_1 and λ_2.[3]

Unsupervised Conditional Image Generation via Rate Reduction. The above representation is learned without class information. In order to facilitate discriminative or generative tasks, it must be highly structured. As we will see via experiments, specific and unique structure indeed emerges naturally in the representations learned using U-CTRL: globally, features of images in the same class tend to be clustered well together and separated from other classes (Figure 2); locally, features around individual samples exhibit approximately piecewise linear low-dimensional structures (Figure 5).

The highly-structured feature distribution also suggests that the learned representation can be very useful for generative purposes. For example, we can organize the sample features into meaningful clusters, and model them with low-dimensional (Gaussian) distributions or subspaces. By sampling from these compact models, we can conditionally regenerate meaningful samples from computed clusters. This is known as *unsupervised conditional image generation* [14].

To cluster features, we exploit the fact that the rate reduction framework (1) is inspired by unsupervised clustering via compression [40], which provides a principled way to find the membership $\boldsymbol{\Pi}$. Concretely, we maximize the same rate reduction objective (1) over $\boldsymbol{\Pi}$, but fix the learned representation $\boldsymbol{Z}$ instead. We simply view the membership $\boldsymbol{\Pi}$ as a nonlinear function of the features $\boldsymbol{Z}$, say $h_{\boldsymbol{\pi}}(\cdot, \xi) : \boldsymbol{Z} \mapsto \boldsymbol{\Pi}$ with parameters ξ. In practice, we model this function with a simple neural network, such as an MLP head right after the output feature $\boldsymbol{z}$. To estimate a "pseudo" membership $\hat{\boldsymbol{\Pi}}$ of the samples, we solve the following optimization problem over $\boldsymbol{\Pi}$:

$$\hat{\boldsymbol{\Pi}} = \arg\max_{\xi} \Delta R(\boldsymbol{Z} | \boldsymbol{\Pi}(\xi)). \tag{10}$$

Experiments in Section 4.2 demonstrate that conditional image generation from clusters produced in this manner result in high-quality images that are highly similar in style.

4. Experiments

We now evaluate the performance of the proposed U-CTRL framework and compare it with representative unsupervised generative and discriminative methods. The first set of experiments (Section 4.1 show that despite being a generative method in nature, U-CTRL can learn discriminative representations competitive with state-of-the-art discriminative methods. The second set (Section 4.2) show that the learned generative representation can significantly boost the performance of unsupervised conditional image generation. Finally, the third set (Section 4.3) study how the advantages that generative represeentations have over discriminative ones.

We conduct experiments on the following datasets: CIFAR-10 [41], CIFAR-100 [42], and Tiny ImageNet [43]. Standard augmentations for self-supervised learning are used across all datasets [1].

We design all experiments to ensure that comparisons against U-CTRL are fair. For all methods that we compare against, we ensure that experiments are conducted with similar model sizes. If code for similar size structure can not be found, we uniformly use ResNet-18 to reproduce results for baselines, which is larger than

[3]Notice that computing the rate reduction terms ΔR for all samples or a batch of samples requires computing the expensive $\log\det$ of large matrices. In practice, from the geometric meaning of ΔR for two vectors, ΔR can be approximated with an ℓ^2 norm or the cosine distance between two vectors.

Method	CIFAR-10	CIFAR-100	Tiny-ImageNet
	Accuracy	Accuracy	Accuracy
GAN based methods			
SSGAN-LA[25]	0.803	0.543	0.344
DAGAN+[44]	0.772	0.519	0.224
ContraD[18]	0.852	0.514	-
VAE based methods			
PATCH-VAE [19]	0.471	0.325	-
β-VAE [26]	0.531	0.315	-
CTRL based methods			
CTRL-Binary[16]	0.599	-	-
U-CTRL (ours)	**0.874**	**0.552**	**0.360**

Table 2: Comparison of classification accuracy on CIFAR-10, CIFAR-100, and Tiny-ImageNet with other generative self-supervised learning methods. U-CTRL is clearly better.

Method	CIFAR-10	CIFAR-100	Tiny-ImageNet
	Accuracy	Accuracy	Accuracy
SIMCLR	0.869	0.545	0.359
MoCoV2	0.872	**0.589**	0.365
BYOL	**0.883**	0.581	**0.371**
U-CTRL (ours)	0.874	0.552	0.360

Table 3: Comparison of classification accuracy on CIFAR-10, CIFAR-100, and Tiny-ImageNet with purely discriminative self-supervised learning methods. U-CTRL is on par with these non-generative methods.

the network used by our method. Details about network architectures and the experimental setting are given in Appendix A. All methods have runned 400 epochs or equivalent iterations (because generative models often count in iteration).

4.1. Discriminative Quality of Learned Representations

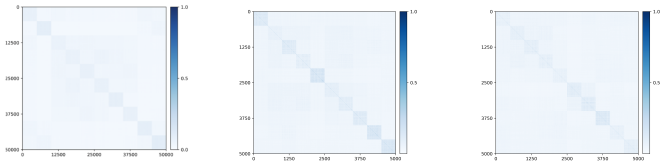
To evaluate the discriminative quality of the learned representations, we follow the standard practice of evaluating the accuracy of a simple linear classifier trained on the learned representation. Table 2 compares our method against SOTA generative self-supervised learning methods, and Table 3 compares our method against SOTA discriminative self-supervised methods. Experimental and training details are given in Appendix A.

Quantitative Comparisons of Classification Performance. From Table 2, we observe that on all chosen datasets, our method achieves substantial improvements compared to existing generative self-supervised learning methods. This includes more complex datasets like CIFAR-100 and Tiny-ImageNet, where we surpass the current SOTA models. From Table 3, our method achieves similar performance compared to SOTA discriminative self-supervised models. These results echo our goal of seeking a more unifed generative and discriminative representations: despite resembling a generative method architecturally, our method still produces highly discriminative representations. In addition, these results lead us to ask a fundamental question: when is incorporating both discriminative and generative properties a whole greater than the sum of its parts, particularly outside of the context of computational efficiency? We provide preliminary answers in Section 4.3.

Qualitative Visualization of Learned Representations. To explain the classification performance of our method, we visualize the incoherence between features learned for the training datasets. Figure 2 shows cosine similarity heatmaps between the learned features, organized by ground-truth class labels. A block-diagonal pattern emerges automatically from U-CTRL training for all three datasets, similar to those observed in features learned in a supervised setting [16]. In this case, however, these blocks emerge and correspond with classes labels despite the absence of any supervision at all.

4.2. Improved Unsupervised Conditional Generation Quality

To evaluate the quality of unsupervised conditional image generation, we measure performance on two axes: cluster quality and image quality. We estimate clusters by optimizing (10), and show results and comparisons



(a) CIFAR-10 (b) CIFAR-100 (c) Tiny ImageNet

Figure 2: Emergence of block-diagonal structures of $|\boldsymbol{Z}^\top \boldsymbol{Z}|$ in the feature space for CIFAR-10 (left), 10 random classes from CIFAR-100 (middle), and 10 random classes from Tiny ImageNet (right).

(a) CIFAR-10 $\boldsymbol{X}$ (b) CIFAR-10 $\hat{\boldsymbol{X}}$

Figure 3: Sample-wise self-consistency: visualization of images $\boldsymbol{X}$ and reconstructed $\hat{\boldsymbol{X}}$ on CIFAR-10 dataset.

Figure 4: Unsupervised conditional image generation from each cluster of CIFAR-10, using U-CTRL. Images from different rows mean generation from different principal components of each cluster.

with both recent and classical methods in Table 4. Training details of our method for the additional MLP head can be found in the Appendix A.

Cluster Quality. We measure normalized mutual information (NMI) and clustering accuracy for cluster quality on CIFAR-10 clustered into 10 classes and CIFAR-100(20), which is clustered into 20 super-classes. From Table 4, we observe that on CIFAR-10, U-CTRL results in an NMI that is almost double that of the existing SOTA on both GAN-based and VAE-based methods, with significantly improved clustering accuracy. Unlike many baselines, we also demonstrate that our method scales to the more challenging CIFAR-100(20) dataset, where it also significantly outperforms alternatives. Our improved clustering quality suggests potential for improving unsupervised conditional image generation, which relies on first finding statistically (and hence visually) meaningful clusters.

Image Quality. We use Frechet Inception Distance (FID) [45] and Inception Score (IS) [46] to measure image quality. From Table 4, it is evident that U-CTRL maintains competitive image quality compared to other methods, measured both by FID and IS. We also compare original images against reconstructed ones in Figure 3, where we see that the original $\boldsymbol{X}$ is very similar to the reconstructed $\hat{\boldsymbol{X}}$; U-CTRL indeed achieves very good sample-wise self-consistency.

Unsupervised Conditional Image Generation. In Figure 4, we visualize images generated from the ten unsupervised clusters from (10). Each block represents one cluster and each row represents one principal component for each cluster. Despite learning and training without labels, the model not only organizes samples into correct clusters, but is also able to preserve statistical diversities within each cluster/class. We can easily recover the diversity within each cluster by computing different principal components and then sample and generate accordingly! More detailed illustrations with more samples is provided in Appendix B.

4.3. Benefits of U-CTRL's Structured Representation

As shown in the previous section, on datasets like CIFAR-10, CIFAR-100, and Tiny-ImageNet, our framework is able to achieve representation quality on par with the best discriminative self-supervised learning methods. A clear advantage of this is computational efficiency; only a single representation needs to be trained for a much broader set of tasks. This subsection aims to provide additional insights on how a unified model can be more beneficial for a broader range of tasks.

Domain Transfer. Regenerating images is demanding on the encoder, which is required to produce a more informative representation than contrastive training would. We hypothesize that the encoder trained with generative task may retain more information about the image and allow the representation to generalize better.

Method	CIFAR-10				CIFAR-100(20)			
	NMI	Accuracy	FID↓	IS↑	NMI	Accuracy	FID↓	IS↑
GAN based methods								
Self-Conditioned GAN [35]	0.333	0.117	18.0	7.7	0.214	0.092	24.1	5.2
SLOGAN [14]	0.340	-	20.6	-	-	-	-	-
VAE based methods								
GMVAE[47]	-	0.247	-	-	-	-	-	-
Variational Clustering	-	0.445	-	-	-	-	-	-
CTRL based methods								
U-CTRL (ours)	**0.658**	**0.799**	**17.4**	**8.1**	**0.374**	**0.433**	**20.1**	**7.7**

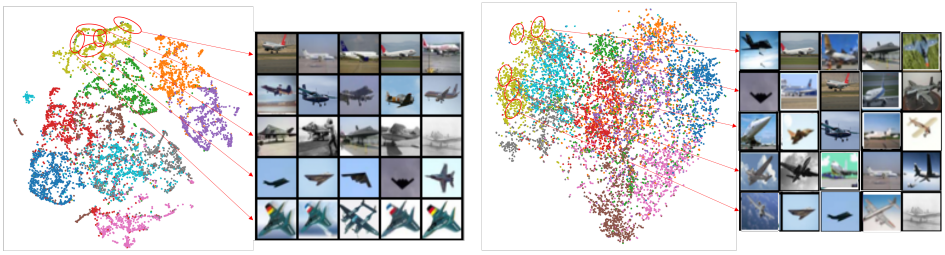
Table 4: Comparison of the quality of UCIG on CIFAR-10 and CIFAR-100(20). Many of the methods compared do not provide code that scales up to CIFAR-100(20), in which case we leave the corresponding table cell blank.

To verify this, we compare the accuracy on CIFAR-100 using models learned from CIFAR-10 in Table 5. When compared to purely discriminative self-supervised learning models, we observe that U-CTRL is 4 percent better than other methods on classification accuracy.

Method	SIMCLR	MoCoV2	BYOL	U-CTRL
Accuracy	0.422	0.436	0.437	0.481

Table 5: Comparing the transfer ability with purely discriminative self-supervised learning methods. All methods are trained unsupervised on CIFAR-10 and tested on CIFAR-100.

Visualization of Emerged Structures. The representations learned by U-CTRL are significantly different from those learned from previous either discriminative and generative methods. To illustrate this, we use t-SNE [48] to visualize the learned representation in 2D. Figure 5 compares the t-SNE of representations learned for CIFAR-10 by U-CTRL and MoCoV2, respectively. It is clear that the representation learned by U-CTRL are much more structured and better organized: classes are more evident, and features within each class form clear piecewise linear structures.



(a) U-CTRL (b) MoCoV2

Figure 5: t-SNE visualizations of learned features of CIFAR-10 with different models.

5. Conclusion and Discussion

In this work, we proposed an unsupervised formulation of the closed-loop transcription framework [16]. We experimentally demonstrate that it is possible to learn a unified representation for both discriminative and generative purposes, resulting in highly structured representations. Further, we show that these two purposes mutually benefit each other in various tasks, e.g., conditional image generation and domain tranfers. Compared to the more specialized representations learned in prior works, our results suggest that such a unified representation has the potential in supporting and benefiting a wider range of new tasks. In future work, we believe the learned representations can be further improved by jointly optimizing the feature representation and feature clusters, as suggested in the original rate reduction paper [15]. Features with high likelihood of belonging to the same cluster can be further linearized and compressed. Due to its unifying nature and the simplicity of the underlying concepts, this new framework may be extended beyond image data, such as sequential or dynamical observations.

References

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

[2] 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*, pages 9729–9738, 2020.

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

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[5] 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*, pages 12310–12320. PMLR, 2021.

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2020.

[7] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[11] Georg B Keller and Thomas D Mrsic-Flogel. Predictive processing: A canonical cortical computation. *Neuron*, 100(2):424–435, October 2018.

[12] Sheena A. Josselyn and Susumu Tonegawa. Memory engrams: Recalling the past and imagining the future. *Science*, 367, 2020.

[13] Gido M Ven, Hava T Siegelmann, Andreas S Tolias, et al. Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, 11(1):1–14, 2020.

[14] Uiwon Hwang, Heeseung Kim, Dahuin Jung, Hyemi Jang, Hyungyu Lee, and Sungroh Yoon. Stein latent optimization for generative adversarial networks. *arXiv preprint arXiv:2106.05319*, 2021.

[15] Kwan Ho Ryan Chan, Yaodong Yu, Chong You, Haozhi Qi, John Wright, and Yi Ma. ReduNet: A white-box deep network from the principle of maximizing rate reduction. *Journal of Machine Learning Research*, 23(114):1–103, 2022. URL `http://jmlr.org/papers/v23/21-0631.html`.

[16] Xili Dai, Shengbang Tong, Mingyang Li, Ziyang Wu, Michael Psenka, Kwan Ho Ryan Chan, Pengyuan Zhai, Yaodong Yu, Xiaojun Yuan, Heung-Yeung Shum, et al. Ctrl: Closed-loop transcription to an ldr via minimaxing rate reduction. *Entropy*, 24(4):456, 2022.

[17] Shengbang Tong, Xili Dai, Ziyang Wu, Mingyang Li, Brent Yi, and Yi Ma. Incremental learning of structured memory via closed-loop transcription. *arXiv:2202.05411*, 2022.

[18] Jongheon Jeong and Jinwoo Shin. Training gans with stronger augmentations via contrastive discriminator. *arXiv preprint arXiv:2103.09742*, 2021.

[19] Gaurav Parmar, Dacheng Li, Kwonjoon Lee, and Zhuowen Tu. Dual contradistinctive generative autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 823–832, 2021.

[20] Zengyi Li, Yubei Chen, Yann LeCun, and Friedrich T Sommer. Neural manifold clustering and embedding. *arXiv preprint arXiv:2201.10000*, 2022.

[21] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

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

[23] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[24] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12154–12163, 2019.

[25] Liang Hou, Huawei Shen, Qi Cao, and Xueqi Cheng. Self-supervised gans with label augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

[26] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *arXiv preprint arXiv:1804.03599*, 2016.

[27] Kamal Gupta, Saurabh Singh, and Abhinav Shrivastava. Patchvae: Learning local latent codes for recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4746–4755, 2020.

[28] William Falcon, Ananya Harsh Jha, Teddy Koker, and Kyunghyun Cho. Aavae: Augmentation-augmented variational autoencoders. *arXiv preprint arXiv:2107.12329*, 2021.

[29] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[30] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.

[31] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020.

[32] Saehoon Kim, Sungwoong Kim, and Juho Lee. Hybrid generative-contrastive representation learning. *arXiv preprint arXiv:2106.06162*, 2021.

[33] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[34] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4610–4617, 2019.

[35] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14286–14295, 2020.

[36] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

[37] Vignesh Prasad, Dipanjan Das, and Brojeshwar Bhowmick. Variational clustering: Leveraging variational autoencoders for image clustering. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2020.

[38] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Advances in Neural Information Processing Systems*, 33:9422–9434, 2020.

[39] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

[40] Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *PAMI*, 2007.

[41] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The CIFAR-10 dataset. *online: http://www.cs.toronto.edu/kriz/cifar.html*, 55, 2014.

[42] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. *arXiv preprint arXiv:1312.6114*, 2009.

[43] 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*, pages 248–255. Ieee, 2009.

[44] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[45] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[46] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[47] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[48] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

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

A. Training Details

A.1. Network Architectures

Table 6, 7 and Figure 6 give details on the network architecture for the decoder and the encoder networks used for experiments. The black rectangle marked with "conv, s=2" means a convlutional layer with stride 2. The orange rectangle marked with "dconv, s=2" means a deconvolutional layer with stride 2. The "x k" besides red frame means we regard these layers in red frame as a block and stack it k times. All α values in Leaky-ReLU (i.e. lReLU) of the encoder are set to 0.2. We set ($nz = 128$, $nc = 3$, $k = 3$) for CIFAR-10, ($nz = 256$, $nc = 3$, $k = 4$) for CIFAR-100, and ($nz = 256$, $nc = 3$, $k = 4$) for Tiny-ImageNet. As a comparison, ResNet-18 contains around 11 million parameters, whereas our encoder only contains between 4 and 6 million parameters depending on the choice of k.

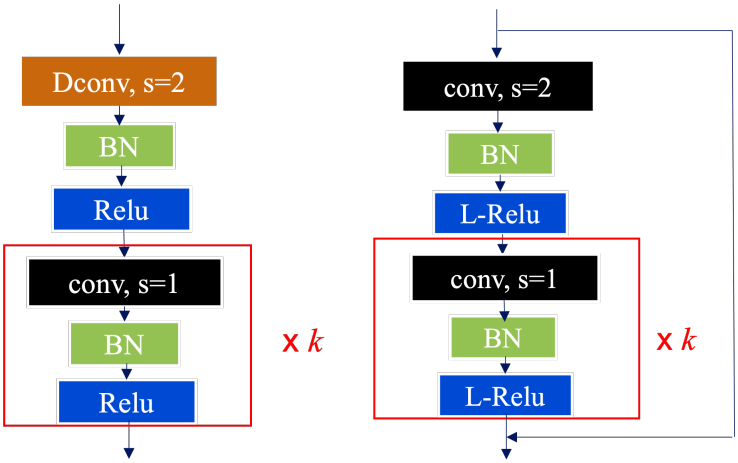
Table 8 gives details of the network architecture for the linear classifier and Table 9 gives details of the network architecture for the additional MLP head used for unsupervised conditional image generation training.

$\boldsymbol{z} \in \mathbb{R}^{1\times1\times nz}$
ResBlockUp. 256
ResBlockUp. 128
ResBlockUp. 64
4×4, stride=2, pad=1 deconv. 1 Tanh

Table 6: Network architecture of the decoder $g(\cdot, \eta)$.

Image $\boldsymbol{x} \in \mathbb{R}^{32\times32\times nc}$
ResBlockDown 64
ResBlockDown 128
ResBlockDown 256
4×4, stride=1, pad=0 conv nz

Table 7: Network architecture of the encoder $f(\cdot, \theta)$.



(a) ResBlock Up architecture (b) ResBlock Down architecture

Figure 6: Architecture of two ResBlock.

$\boldsymbol{z} \in \mathbb{R}^{1\times1\times nz}$
Linear(nz, number of class)

Table 8: Network architecture of the linear classifier.

$\boldsymbol{z} \in \mathbb{R}^{1\times1\times nz}$
Linear(nz, nz) ReLU
Linear(nz, number of clusters)

Table 9: Network architecture of the MLP head for unsupervised conditional image generation

A.2. Optimization

For all experiments, we use Adam [49] as our optimizer, with hyperparameters $\beta_1 = 0.5, \beta_2 = 0.999$. The learning rate is set to be 0.0001. We choose $\epsilon^2 = 0.2$. For all experiments, we adopt augmentation from SimCLR [1].

For CIFAR-10, CIFAR-100, and Tiny ImageNet, we train our framework with a batch size of 1024 over 20,000 iterations. All experiments are conducted with at most 4 RTX 3090 GPUs. Methods that are compared against in Table 3 are trained with the batch size of 256, because Chen et al. [1] observe that purely discriminative methods tend to perform better with smaller batch sizes. Table 2 methods have used their optimal parameters in their github code.

For training of the MLP head for unsupervised conditional image generation(10), we again use Adam [49] as our optimizer with hyperparameters $\beta_1 = 0.5, \beta_2 = 0.999$. We choose the learning rate to be 0.0001 and ϵ^2 as 0.2, with batch size 1024 over 5000 iterations.

For training of the linear classifier, we use Adam [49] as our optimizer with hyperparameters $\beta_1 = 0.5, \beta_2 = 0.999$. We choose learning rate to be 0.0001, with batch size 1024 over 5000 iterations.

B. Additional Unsupervised Clustering and Generation Results

B.1. Cluster Reconstruction

In this subsection, we visualize the reconstruction of ten clusters that are predicted and generated by U-CTRL on the CIFAR-10 training set. Each block in Figure 7 contains both a random sample of reconstructed data in a cluster and the total number of samples within it. Note that CIFAR-10 contains 50,000 training samples, split across 10 classes. As we see in Figure 7, the number of samples in each cluster are very close to 5,000, with the largest deviator (cluster 9) containing 3,942 samples. Without any cues, one can easily identify correspond each unsupervised cluster with a CIFAR-10 class. For a class like 'bird', we observe that the model is able to group images of standing birds, flying birds, and bird heads, despite their visual differences.

B.2. Unsupervised Conditional Image Generation

Building on U-CTRL's ability to cluster CIFAR-10 samples, we demonstrate the model's ability to perform unsupervised conditional image generation in Figure 8. In contrast to reconstruction, where images are regenerated from features corresponding to real samples, we generate images based on the feature sampling technique proposed in [16]. From these results, we observe that the U-CTRL framework maintains in-cluster diversity, and that the diversity can be recovered and visualized via simple principal component analysis.

(a) Cluster 1 (b) Cluster 2

(c) Cluster 3 (d) Cluster 4

(e) Cluster 5 (f) Cluster 6

(g) Cluster 7 (h) Cluster 8

(i) Cluster 9 (j) Cluster 10

Figure 7: More result on the reconstruction of clusters in CIFAR-10.

C. Ablation Studies

C.1. The importance of each term in U-CTRL formulation

In this section, we study the significance of the sample-wise constraints and extra rate distortion term in the formulation 7. Table 10 presents the following objectives that we study:

- Objective I is the constrained U-CTRL maximin 7.
- Objective II is the constrained maximin without the augmentation compression constraint 6.
- Objective III is the constrained maximin without the sample-wise self-consistency constraint 5.
- Objective IV is the constrained maximin without the extra rate distortion term.
- Objective V is the U-CTRL without the augmentation compression constraint and sample-wise self-consistency constraint.
- Objective VI is the CTRL-Binary maximin formulation 4.

Table 11 shows the result of a linear probe for representations trained using each objective on CIFAR-10. From the table, it is evident that both constraints and the rate distortion term are pivotal to the success of our framework.

Objective I:	$\max_\theta \min_\eta R(\boldsymbol{Z}) + \Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}})$ s.t. $\sum_{i \in N} \Delta R(\boldsymbol{z}^i, \hat{\boldsymbol{z}}^i) = 0$, and $\sum_{i \in N} \Delta R(\boldsymbol{z}^i, \boldsymbol{z}_a^i) = 0$
Objective II:	$\max_\theta \min_\eta R(\boldsymbol{Z}) + \Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}})$ s.t. $\sum_{i \in N} \Delta R(\boldsymbol{z}^i, \hat{\boldsymbol{z}}^i) = 0$
Objective III:	$\max_\theta \min_\eta R(\boldsymbol{Z}) + \Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}})$ s.t. $\sum_{i \in N} \Delta R(\boldsymbol{z}^i, \boldsymbol{z}_a^i) = 0$
Objective IV:	$\max_\theta \min_\eta \Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}})$ s.t. $\sum_{i \in N} \Delta R(\boldsymbol{z}^i, \hat{\boldsymbol{z}}^i) = 0$, and $\sum_{i \in N} \Delta R(\boldsymbol{z}^i, \boldsymbol{z}_a^i) = 0$
Objective V:	$\max_\theta \min_\eta R(\boldsymbol{Z}) + \Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}})$
Objective VI:	$\max_\theta \min_\eta \Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}})$

Table 10: Five different objective functions for U-CTRL.

Method	Objective I	Objective II	Objective III	Objective IV	Objective V	Objective VI
Accuracy	0.874	0.578	0.644	0.522	0.633	0.599

Table 11: Ablation study on the significance of different terms in U-CTRL.

C.2. The importance of MCR^2 in U-CTRL formulation

In this section, we verify the significance of MCR^2 term $\Delta R(\boldsymbol{Z}, \hat{\boldsymbol{Z}})$ in our method. We do ablation study on CIFAR-10 with the same network and training condition. If we take away MCR^2 from our formulation, it changes (11). For simplicity, we call it U-CTRL-noMCR^2

$$\begin{aligned} \max_{\theta} \min_{\eta} \quad & R(\boldsymbol{Z}) \\ \text{subject to} \quad & \sum_{i \in N} \Delta R(\boldsymbol{z}^i, \hat{\boldsymbol{z}}^i) = 0, \quad \text{and} \quad \sum_{i \in N} \Delta R(\boldsymbol{z}^i, \boldsymbol{z}_a^i) = 0. \end{aligned} \tag{11}$$

Table 12 shows that U-CTRL without the MCR^2 not only learns worse representation but also generalizes worse to out of distribution data. Figure 9 visualizes the reconstructed $\hat{\boldsymbol{X}}$ by U-CTRL-noMCR^2. It is clear from the image figure that without the MCR^2, the decoder fails to reconstruct high-quality images.

	Accuracy on CIFAR-10	Transfer Accuracy on CIFAR-100
U-CTRL	0.874	0.481
U-CTRL-noMCR^2	0.836	0.418

Table 12: Ablation study on the significance of MCR^2 in U-CTRL.

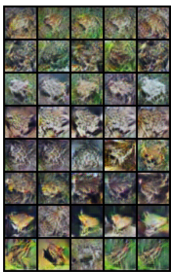
It follows our discussion in the introduction that discriminative tasks and generative tasks together learn feature that benifits each other.

D. Random Seed Sensitivity

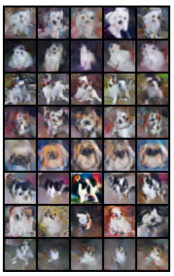
In this section, we verify the stability of our method against different random seeds. We report in Table 13 the accuracy of U-CTRL on CIFAR-10 with different seeds. We observe that the choice of seed has very little impact on performance.

Random Seed	1	5	10	15	100
Accuracy	0.874	0.876	0.870	0.874	0.871

Table 13: Ablation study on varying random seeds.

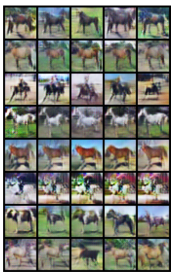


(a) Cluster 1

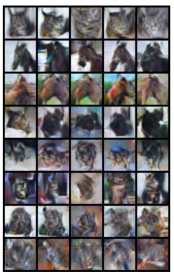


(b) Cluster 2

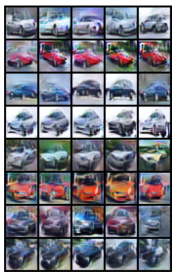
(c) Cluster 3



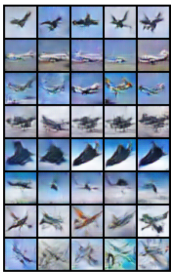
(d) Cluster 4



(e) Cluster 5



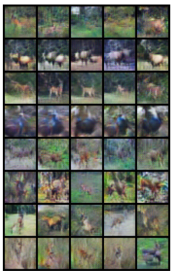
(f) Cluster 6



(g) Cluster 7

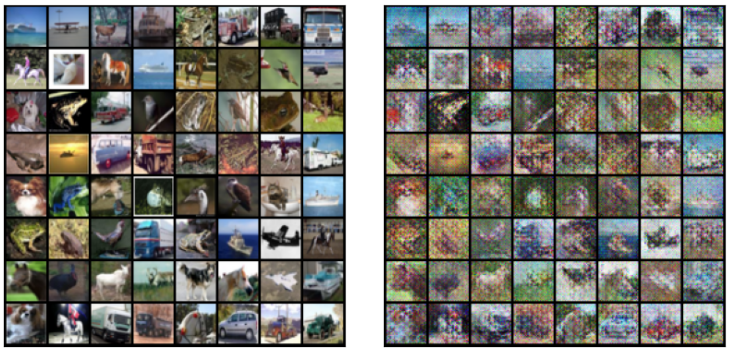
(h) Cluster 8

(i) Cluster 9



(j) Cluster 10

Figure 8: Unsupervised conditional image generation on CIFAR-10. Each block represents a cluster, within which each row represents one principal component direction in the cluster, and samples along each row represent different noises applied in that principal direction.



(a) CIFAR-10 $\boldsymbol{X}$ (b) CIFAR-10 $\hat{\boldsymbol{X}}$

Figure 9: Visualization of images trained by U-CTRL-noMCR^2: $\boldsymbol{X}$ and reconstructed $\hat{\boldsymbol{X}}$ on CIFAR-10 dataset.