Learning Robust Representations via Nuisance-extended Information Bottleneck

Jongheon Jeong Sihyun Yu Hankook Lee Jinwoo Shin
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, South Korea
{jongheonj, sihyun.yu, hankook.lee, jinwoos}@kaist.ac.kr

Abstract

The information bottleneck *(IB) principle is one of natural approaches to obtain a succinct representation* $\mathbf{x} \to \mathbf{z}$ *for a given downstream task* $\mathbf{x} \to \mathbf{y}$*: namely, it finds* $\mathbf{z}$ *that (a) maximizes the (task-relevant) mutual information* $I(\mathbf{z}; \mathbf{y})$*, while (b) minimizing* $I(\mathbf{x}; \mathbf{z})$ *to constrain the capacity of* $\mathbf{z}$ *for better generalization. In practical scenarios where the training data is limited, however, the IB objective may not be able to prevent* $\mathbf{z}$ *from co-adapting on so-called "shortcut" signal, i.e., features only in training data those are predictive-yet-compressible enough. They are typically from biases in data acquisition, and less generalizable under new (but still semantically-aligned) environments. To bypass such a failure mode, we extend the standard framework of IB to also model the* nuisance information *with respect to* $\mathbf{z}$*, namely* $\mathbf{z}_n$*, so that* $(\mathbf{z}, \mathbf{z}_n)$ *can reconstruct* $\mathbf{x}$*: by minimizing* $I(\mathbf{z}_n; \mathbf{y})$ *as well as the IB objective here,* $\mathbf{z}$ *can now encode more diverse* $\mathbf{y}$*-related signal in* $\mathbf{x}$*, while disentangling the remainder information from* $\mathbf{z}$*. Our experimental results show that the representation learned from our proposed training consistently improves various notions of robustness over the standard VIB training without relying on data augmentations, e.g., novelty detection and corruption robustness.*

1. Introduction

Generally speaking, a neural network model, say f, is a parametric mapping of a given random variable $\mathbf{x}$ into its representation $\mathbf{z} := f(\mathbf{x})$, that encodes useful features in $\mathbf{x}$ to predict a target random variable $\mathbf{y}$ so that a simpler (*e.g.*, linear) mapping can recover $\mathbf{y}$ from $\mathbf{z}$: in other words, a "good" representation $\mathbf{z}$ should keep information of $\mathbf{x}$ that is correlated with $\mathbf{y}$, while preventing $\mathbf{z}$ from being too complex. The *information bottleneck* (IB) principle [97, 98] is a simple and natural implementation, which sets the *mutual information* $I(\mathbf{x}; \mathbf{z})$ as complexity measure of $\mathbf{z}$:

$$\max_f R_{\texttt{IB}}(f), \quad \text{for} \quad R_{\texttt{IB}}(f) := I(\mathbf{z}; \mathbf{y}) - \beta I(\mathbf{x}; \mathbf{z}), \quad (1)$$

where $\beta \geq 0$ controls the capacity constraint which ensures $I(\mathbf{x}; \mathbf{z}) \leq I_\beta$ for some (implicitly defined) I_β.

That being said, the brittleness of neural networks for out-of-distribution samples can still persist even with the IB objective (1): in other words, a "good" model f from the objective can work poorly under a certain *distribution shift* in $\mathbf{x}$, say $\hat{\mathbf{x}}$, so that $I(\mathbf{x}; \mathbf{y}) = I(f(\mathbf{x}); \mathbf{y}) \gg I(f(\hat{\mathbf{x}}); \mathbf{y})$. In practice, this can occur especially when the (hard-to-compute) mutual information terms in (1) are approximated based on limited, and potentially biased data: for example, many well-curated datasets commonly used in research [62, 88] are likely to be processed prior to release for quality control, *e.g.*, by filtering out some severely corrupted samples from its original collection. Such a bias can make the computation of $I(\mathbf{z}; \mathbf{y})$ to be also biased, *i.e.*, toward over-estimating a "shortcut" signal [24] in the data that is not generalizable for $\hat{\mathbf{x}}$. Even worse, by jointly minimizing $I(\mathbf{x}; \mathbf{z})$ in (1), it can further compress out other useful signal in $\mathbf{x}$ if the shortcuts are already predictive enough.

Contribution. In this paper, we rethink the implementation of the information bottleneck (IB) principle under presence of distribution shifts. In particular, we argue that a "robust" representation $\mathbf{z}$ should always encode *every* signal in $\mathbf{x}$ that is correlated with $\mathbf{y}$, rather than extracting only a few shortcuts; the capacity constraint in IB (1) can still be applied for the *nuisance* information which is not related to predict $\mathbf{y}$ at all. We propose a practical design of this framework by incorporating a *nuisance representation* $\mathbf{z}_n$ alongside $\mathbf{z}$ of the standard IB framework so that $(\mathbf{z}, \mathbf{z}_n)$ can reconstruct $\mathbf{x}$. This results in a novel synthesis of *adversarial autoencoder* [77] and *variational information bottleneck* [1] into a single framework.

At a high level, our method can be viewed as a new approach of improving the robustness of discriminative classifiers by incorporating a generative model. For example, [66] and [29] use a simple Gaussian mixture model of low expressive power and an energy-based model of training instability for the purpose, respectively. Our approach of incorporating autoencoder-based models takes the best of two worlds; it enables (a) stable training, while (b) attaining the high

expressive generative performances. Regarding the literature of nuisance modeling [43, 46, 84], on the other hand, our work is the first to the best of our knowledge on exploring and designing a successful VIB-based framework to improve multiple recent safety measures, *e.g.*, both in corruption/adversarial robustness and out-of-distribution detection, as well as establishing new practices to scale-up the previous approaches that were mostly limited in MNIST-scale.[1]

2. Nuisance-extended IB

Notations. Given two random variables $\mathbf{x} \in \mathcal{X}$, the input, and $\mathbf{y} \in \mathcal{Y}$, the target, we consider a general problem of *representation learning* [1, 7, 19, 21, 60, 83], where the goal is to find a mapping (or an *encoder*) $f : \mathcal{X} \to \mathcal{Z}$ from data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$[2] so that $\mathbf{z} := f(\mathbf{x})$, the representation, can predict $\mathbf{y}$ with a simper (*e.g.*, linear) mapping. We assume that the encoder f is parametrized by a neural network, and the mapping is *stochastic* to adopt an information theoretic view of neural networks [98], *i.e.*, the encoder output is a random variable defined as $p_f(\mathbf{z}|\mathbf{x})$ rather than a constant. In practice, such a modeling can be done through the *reparametrization trick* [58], *i.e.*, by allowing an independent random variable $\boldsymbol{\epsilon}$ to the (deterministic) mapping f as an additional input, namely $\mathbf{z} := f(\mathbf{x}, \boldsymbol{\epsilon})$. For example, a popular design of *Gaussian decoder* parametrizes f by:

$$f(\mathbf{x}, \boldsymbol{\epsilon}) := f^{\mu}(\mathbf{x}) + \boldsymbol{\epsilon} \cdot f^{\sigma}(\mathbf{x}), \tag{2}$$

so that the deterministic f^{μ} and f^{σ} can still be learned through gradient-based optimization.

Nuisance-extended IB. The standard information bottleneck (IB) objective (1) obtains a representation $\mathbf{z} := f(\mathbf{x})$ on premise that the future inputs will be also from the data generating distribution $p_d(\mathbf{x}, \mathbf{y})$. In this paper, we aim to extend the IB objective under assumption that the input $\mathbf{x}$ can possibly be corrupted through an *unknown* noisy channel in the future, say $\mathbf{x} \to \hat{\mathbf{x}}$, while $\hat{\mathbf{x}}$ still preserves the *semantics* of $\mathbf{x}$ with respect to $\mathbf{y}$: in other words, we assume $I(\mathbf{x}; \mathbf{y}) = I(\hat{\mathbf{x}}; \mathbf{y}) > 0$. Intuitively, one can imagine a scenario that a given input $\mathbf{x}$ contains multiple signals that each is already highly correlated with $\mathbf{y}$, *i.e.*, filtering out the remainder from $\mathbf{x}$ does not affect its mutual information with $\mathbf{y}$. It may or may not be surprising that such signals are quite prevalent in practical deep neural networks, *e.g.*, [41] empirically observe that adversarial perturbations [27, 93] crafted from a given neural network are sufficient for the model to perform accurate classification.

In the context of IB framework, where the goal is to obtain a succinct encoder f, it is now reasonable to presume that the noisy channel $\hat{\mathbf{x}}$ acts like an *adversary*, *i.e.*, it minimizes:

$$\min_{\hat{\mathbf{x}}} I(\hat{\mathbf{z}} := f(\hat{\mathbf{x}}); \mathbf{y}) \;\text{ subject to }\; I(\mathbf{x}; \mathbf{y}) = I(\hat{\mathbf{x}}; \mathbf{y}), \tag{3}$$

given that one has no information on how the channel would behave *a priori*. This minimax optimization thus would require f to extract *every* signal in $\mathbf{x}$ whenever it is highly correlated with $\mathbf{y}$, to avoid the case when $\hat{\mathbf{x}}$ filter out all the signal except one that f has missed. We notice that, nevertheless, directly optimizing (3) with respect to $\hat{\mathbf{x}}$ is computationally infeasible in practice, considering that (a) it is in many cases an unconstrained optimization in a high-dimensional $\mathcal{X}$, (b) with a constraint on (hard-to-compute) mutual information.

In this paper, to make sure that f still exhibits the "adversarial" behavior without (3), we propose to let f to model the *nuisance representation* $\mathbf{z}_n$ as well as $\mathbf{z}$: specifically, $\mathbf{z}_n$ aims to model the "remainder" information from $\mathbf{z}$ needed to reconstruct $\mathbf{x}$, *i.e.*, it maximizes $I(\mathbf{x}; \mathbf{z}, \mathbf{z}_n)$, while compressing out information that is correlated with $\mathbf{y}$, *i.e.*, it also minimizes $I(\mathbf{z}_n; \mathbf{y})$: therefore, every information that is correlated with $\mathbf{y}$ should be encoded into $\mathbf{z}$ in a complementary manner. Here, we remark that now the role of the capacity constraint in (1) becomes even more important: not only for regularizing $\mathbf{z}$ to attain simpler representation, it additionally penalizes $\mathbf{z}_n$ from pushing out unnecessary information to predict $\mathbf{y}$ into $\mathbf{z}$, making the objective competitive again between $\mathbf{z}$ and $\mathbf{z}_n$ as like in (3). Combined with the original IB objective (1), we get:

$$\max_{f} R_{\texttt{NIB}}(f) := R_{\texttt{IB}}(f) - I(\mathbf{z}_n; \mathbf{y}) + \alpha I(\mathbf{x}; \mathbf{z}, \mathbf{z}_n), \tag{4}$$

where $\alpha \geq 0$. The proposed objective, *nuisance-extended IB* (NIB), can be viewed as a regularized form of IB by introducing $\mathbf{z}_n$. This attains the optimal when $I(\mathbf{x}; \mathbf{z}, \mathbf{z}_n)$ and $I(\mathbf{z}_n; \mathbf{y})$ in (4) are maximized and minimized, respectively, *i.e.*, with the conditions of $H(\mathbf{x}|\mathbf{z}, \mathbf{z}_n) = 0$ and $I(\mathbf{z}_n; \mathbf{y}) = 0$. The following observation highlight that having these conditions, additionally with the independence $\mathbf{z} \perp \mathbf{z}_n$, leads f that can recover the original information of $I(\mathbf{x}; \mathbf{y})$ from the noisy channel $I(\hat{\mathbf{z}}; \mathbf{y})$:

Lemma 1. *Let $\mathbf{x} \in \mathcal{X}$, and $\mathbf{y} \in \mathcal{Y}$ be random variables, $\hat{\mathbf{x}}$ be a noisy observation of $\mathbf{x}$ with $I(\mathbf{x}; \mathbf{y}) = I(\hat{\mathbf{x}}; \mathbf{y})$. Given that a representation $[\hat{\mathbf{z}}, \hat{\mathbf{z}}_n] := f(\hat{\mathbf{x}})$ of $\hat{\mathbf{x}}$ satisfies (a) $H(\hat{\mathbf{x}}|\hat{\mathbf{z}}, \hat{\mathbf{z}}_n) = 0$, (b) $I(\hat{\mathbf{z}}_n; \mathbf{y}) = 0$, and (c) $\hat{\mathbf{z}} \perp \hat{\mathbf{z}}_n$, it holds $I(\hat{\mathbf{z}}; \mathbf{y}) = I(\mathbf{x}; \mathbf{y})$.*

A practical design. Based on the NIB objective defined in (4) and Lemma 1, we design a practical training objective to implement the proposed framework. Here, we present a simple instantiation of NIB by approximating it with a synthesis of *adversarial autoencoder* [21] and *variational information bottleneck* (VIB) [1], calling it *nuisance-extended informa-*

[1] We provide a more extensive and broader discussions on related works in Appendix E.

[2] Although we focus here on the setup of *supervised learning*, the framework itself in general does not rule out more general scenarios, *e.g.*, when the target $\mathbf{y}$ can be *self-supervised* from $\mathbf{x}$ [12, 83].

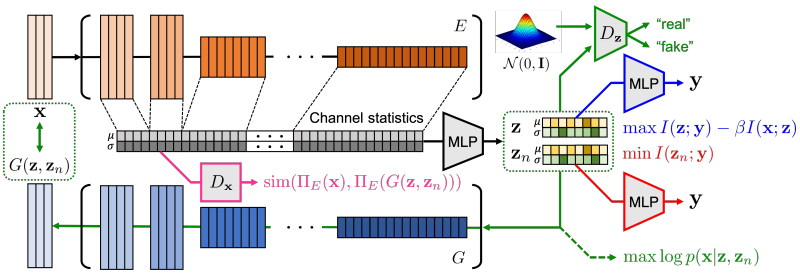


Figure 1. An overview of our proposed framework, *nuisance-extended variational information bottleneck* (NVIB). Overall, the training incorporates adversarial autoencoder into the variational information bottleneck by introducing a *nuisance* $\mathbf{z}_n$ with respect to $\mathbf{y}$ in representation learning. We also propose an *adversarial similarity* based reconstruction to further accelerate the training.

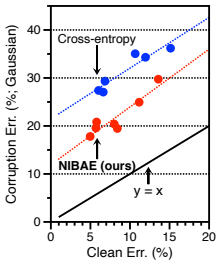
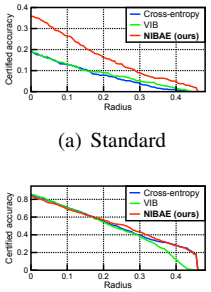


Figure 2. Comparison of training methods in clean vs. corruption errors (against Gaussian) on ViT-S/4 architecture.



(a) Standard

(b) BatchNorm-adapted

Figure 3. Comparison of certified robust accuracy at r on CIFAR-10.

tion bottleneck autoencoder (NIBAE).[3] Figure 1 illustrates an overview of our framework.

Overall, Lemma 1 states that a robust encoder f demands for a "good" nuisance model that achieves generalization on $\hat{\mathbf{z}}$ in three aspects: (a) a *good reconstruction*, (b) *nuisance-ness*, and (c) the *independence between* $\mathbf{z}$ *and* $\mathbf{z}_n$. To model these behaviors, we consider a decoder model decoder $g : \mathcal{Z} \to \mathcal{X}$ as well as the encoder $f : \mathcal{X} \to \mathcal{Z}$, and adopt the following practical training objectives which incorporates an autoencoder-based loss and two adversarial losses originally defined for *generative adversarial networks* (GANs) [26]:

(a) We first pose the standard mean-squared-error based reconstruction loss, which assumes that the decoder output follows Gaussian distribution of constant variance: *i.e.*,

$$L_{\texttt{recon}} := -\log p(\mathbf{x}|\mathbf{z},\mathbf{z}_n) = \frac{1}{2}\|\mathbf{x} - g(\mathbf{z},\mathbf{z}_n)\|_2^2. \quad (5)$$

(b) To force the nuisance-ness of $\mathbf{z}_n$ with respect to $\mathbf{y}$, on the other hand, we approximate $p(\mathbf{y}|\mathbf{z}_n)$ variationally with an MLP, say $q_n(\mathbf{y}|\mathbf{z}_n)$, and perform an adversarial training:

$$L_{\texttt{nuis}} := \mathbb{E}_\mathbf{x}[\mathbb{CE}(q_n^*(\mathbf{z}_n), \tfrac{\mathbf{1}}{|\mathcal{Y}|})], \quad (6)$$

where $q_n^* := \min_{q_n} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbb{CE}(q_n(\mathbf{z}_n),\mathbf{y})]$, and $\mathbb{CE}$ denotes the cross entropy loss. Here, it optimizes the cross-entropy towards the "uniform" distribution in $\mathcal{Y}$.

(c) To induce the independence between $\mathbf{z}$ and $\mathbf{z}_n$, we assume that the joint prior of $\mathbf{z}$ and $\mathbf{z}_n$ is the isotropic Gaussian, *i.e.*, $p(\mathbf{z},\mathbf{z}_n) \sim \mathcal{N}(0,I)$, and performs a GAN:

$$L_{\texttt{ind}} := \max_{q_\mathbf{z}} \mathbb{E}_\mathbf{x}[\log(q_\mathbf{z}(E(\mathbf{x})))] \quad (7)$$

$$+ \mathbb{E}_{\mathbf{z},\mathbf{z}_n\sim\mathcal{N}(0,I)}[\log(1 - q_\mathbf{z}(\mathbf{z},\mathbf{z}_n))], \quad (8)$$

where $q_\mathbf{z}$ is an MLP discriminating $[\mathbf{z},\mathbf{z}_n]$ from $\mathcal{N}(0,I)$.

[3] We also design an architecture for NIBAE in Appendix F.

Lastly, to approximate the original IB objective $R_{\texttt{IB}}(f)$ in NIB (4), we instead maximize the *variational information bottleneck* (VIB) [1] objetive $L^\beta_{\texttt{VIB}}$, that can provide a lower bound on $R_{\texttt{IB}}$.[4] Specifically, it makes variatonal approximations of: (a) $p(\mathbf{y}|\mathbf{z})$ by a (parametrized) "decoder" neural network $q(\mathbf{y}|\mathbf{z})$, and (b) $p(\mathbf{z})$ by an "easier" distribution $r(\mathbf{z})$, *e.g.*, isotropic Gaussian $\mathcal{N}(\mathbf{z}|0,I)$. Recalling that we assume a Gaussian decoder (2) for $f(\mathbf{x},\boldsymbol{\epsilon})$, we have:

$$L^\beta_{\texttt{VIB}} := \mathbb{E}_\boldsymbol{\epsilon}[-\log q(\mathbf{y}|f(\mathbf{x},\boldsymbol{\epsilon}))] + \beta\,\mathrm{KL}\,(p(\mathbf{z}|\mathbf{x})\|r(\mathbf{z})). \quad (9)$$

Overall training objective. Combining the proposed objectives as well as the original VIB loss, $L^\beta_{\texttt{VIB}}$ (9) leads us to the final objective. Although combining multiple losses in practice may introduce additional hyperparameters, we found most of the proposed losses can be added without scaling except for the reconstruction loss $L_{\texttt{recon}}$ and the β in the original VIB loss. Hence, we get:

$$L_{\texttt{NIBAE}} := L^\beta_{\texttt{VIB}} + \alpha \cdot L_{\texttt{recon}} + L_{\texttt{nuis}} + L_{\texttt{ind}} \quad (10)$$

Algorithm 1 (in Appendix A) summarizes the overall procedure of NIBAE training.[5]

3. Experiments

We verify the effectiveness of our proposed NIBAE training for various aspects of out-of-distribution generalization compared to the standard VIB: specifically, we cover (a) novelty detection (Section 3.1 and Appendix G.1), (b) corruption robustness (Section 3.2 and Appendix G.2), (c) adversarial robustness (Appendix G.3) tasks which all have been challenging without assuming task-specific priors [35, 37, 76]. We also present evaluations on the effectiveness of our proposed components in the context of unconditional generative

[4] A more detailed description on the VIB framework (as well as on GAN) can be found in Appendix E.2.

[5] Upon the NIBAE loss (10), we further propose an auxiliary loss for better generation quality, namely *adversarial similarity loss*, in Appendix F.

Table 1. Comparison of OOD detection performances on OBJECTS [105], which considers CIFAR-10-C and ImageNet-10 as additional in-distribution upon CIFAR-10 (training). Bold and underline denote the best and runner-up results, respectively.

Method	AUROC (↑) / AUPR (↑) / FPR95 (↓)	
Cross-entropy	$\max_y p(y\|x)$ [34]	72.27 / 81.47 / 88.63
	ODIN [68]	72.23 / 78.76 / 76.98
	Energy-based [70]	67.01 / 74.75 / 80.96
	Mahalanobis [66]	74.38 / 82.25 / 80.07
	SEM [105]	78.42 / 89.33 / 89.87
	$\mathbf{\log \mathrm{Dir}_{0.05}(y)}$	77.28 / 86.28 / 83.64
VIB [1]	$\max_y p(y\|x)$ [34]	76.84 / 87.46 / 84.41
	$\mathbf{\log \mathrm{Dir}_{0.05}(y)}$	79.91 / 89.48 / 78.05
NIBAE (Ours)	$\max_y p(y\|x)$ [34]	76.56 / 86.75 / 84.20
	$\mathbf{\log \mathrm{Dir}_{0.05}(y)}$	<u>82.38</u> / <u>91.61</u> / <u>73.94</u>
	$\mathbf{+ \log \mathcal{N}(z_n; 0, I)}$	**87.21 / 93.83 / 61.34**

modeling in Appendix I. We provide an ablation study in Appendix C for a component-wise analysis on the method. The full details on the experiments, *e.g.*, datasets, training details, and hyperparameters, can be found in Appendix B.

3.1. Out-of-distribution detection

We first show that our NIBAE model can be a good detector for *out-of-distribution samples* (OODs), *i.e.*, to solve the *novelty detection* task: in general, the task is defined by a binary classification problem that aims to discriminates novel samples from in-distribution samples. We propose two score functions: (a) the Dirichlet score $\log \mathrm{Dir}_{0.05}(y)$ which is applicable for other models, and (b) the nuisance score $\log \mathcal{N}(z_n; 0, I)$ that is unique to NIBAE models (see Appendix G.1 for the details). We consider two evaluation benchmarks and compare ResNet-18 [30] models trained on CIFAR-10: (a) the *"standard"* benchmark, that has been actively adopted in the literature [34, 66, 68] assumes the test set of CIFAR-10 as in-distribution and measures the detection performance of other independent datasets;[6] (b) the *OBJECTS* benchmark, recently proposed in [105], further extend the standard benchmark on CIFAR-10 to also consider "near" in-distribution samples in OOD evaluation. Specifically, OBJECTS assumes CIFAR-10-C [32] and ImageNet-10 [88] as in-distribution in test-time as well as CIFAR-10, making the detection task much more challenging [105].

The results are reported in 1 for the OBJECTS benchmark: overall, we confirm that the score function combining the information of $\mathbf{z}_n$ and $\mathbf{y}$ of NIBAE significantly improve novelty detection in a complementary manner over strong baselines, showing the effectiveness of modeling nuisance. For example, Regarding Table 1, on the other hand, our method of NIBAE shows even more significant improvements here: *e.g.*, NIBAE improves the previous best AUROC (of Mahalanobis [66]) on OBJECTS *vs.* MNIST from $77.04 \rightarrow 92.43$. This shows that both representation and score obtained from NIBAE help to better discriminate in- *vs.* out-of-distribution samples in a more semantic sense compared to previous detection methods.

[6] The results on the "standard" benchmark are reported in Appendix G.1.

Table 2. Comparison of test error rates (%) on CIFAR-10 and its variants, namely CIFAR-10-C [32], CIFAR-10.1 [86], 10.2 [75], and CINIC [17]. We use ViT-S/4 for this experiment. Bold and underline indicate the best and runner-up results, respectively.

Method	C-10	C-10-C	C-10.1	C-10.2	CINIC
Cross-entropy	6.08	16.0	13.4	18.3	23.7
VIB [1]	5.98	15.2	13.6	16.8	23.6
AugMix [37]	6.52	15.1	14.2	17.2	24.2
PixMix [38]	5.43	<u>10.3</u>	13.1	16.6	23.2
NVIB (Ours)	<u>4.97</u>	12.3	<u>11.6</u>	<u>15.5</u>	<u>22.2</u>
+ AugMix [37]	5.35	12.0	12.5	15.8	22.6
+ PixMix [38]	**4.67**	**8.08**	**10.4**	**14.8**	**22.1**

3.2. Robustness against natural corruptions

Next, we evaluate the corruption robustness of our method, namely, the generalization ability of a representation in the situation that the given input can be distorted with natural corruptions (*e.g.*, fog, brightness, etc.) those are still semantic to humans. To this end, we consider (a) CIFAR-10/100-C [33], a corrupted version of CIFAR-10/100 simulating 15 common corruptions in 5 severity levels, respectively, as well as (b) CIFAR-10.1 [86], CIFAR-10.2 [75], and CINIC-10 [17], *i.e.*, three re-generations of the CIFAR-10 test set for the purpose of measuring generalization. From these experiments, we aim to verify that NIBAE can improve *effective robustness* [95] without a domain-specific prior knowledge: in particular, we are interested in improving corruption accuracy under control of the similar level of clean accuracy, which has been challenging given the strong (linear) correlation observed between them across models [31, 79, 95]. We test two encoder architectures, namely ResNet-18 [30] and ViT-S [22, 99], to also investigate the effect of architectures in NIBAE.[7]

In Table 2, we observe that NIBAE significantly and consistently improves corruption errors upon VIB, and these gains are strong even compared with state-of-the-art methods: *e.g.*, NIBAE can solely outperform a strong baseline of AugMix [37]. Although a more recent method of PixMix [38] could achieve a lower corruption error by utilizing extra (pattern-like) data, we remark that (a) NIBAE also benefit from PixMix (*i.e.*, the extra data) as given in "NIBAE + PixMix", and (b) the generalization capability of NIBAE is better than PixMix on CIFAR-10.1, 10.2 and CINIC-10, *i.e.*, in beyond common corruptions, by less relying on domain-specific data. Figure 2 compares the linear trends made by Cross-entropy and NIBAE across different data augmentations and hyperparameters, confirming that NIBAE exhibits a better operating points.

[7] We report the ResNet-18 results on CIFAR-10/100-C in Appendix G.2.

References

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017. 1, 2, 3, 4, 13, 16, 17

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016. 12

[3] Jyoti Aneja, Alex Schwing, Jan Kautz, and Arash Vahdat. A contrastive learning approach for training variational autoencoder priors. *Advances in Neural Information Processing Systems*, 34, 2021. 19

[4] Lynton Ardizzone, Radek Mackowiak, Carsten Rother, and Ullrich Köthe. Training normalizing flows with the information bottleneck for competitive generative classification. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7828–7840. Curran Associates, Inc., 2020. 13

[5] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283. PMLR, 10–15 Jul 2018. 12

[6] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019. 13

[7] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995. 2

[8] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Better plain ViT baselines for ImageNet-1k. *arXiv preprint arXiv:2205.01580*, 2022. 11

[9] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness, 2019. 12

[10] Matthew Chalk, Olivier Marre, and Gasper Tkacik. Relevant sparse codes with variational information bottleneck. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 13

[11] Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems*, 32, 2019. 13

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2020. 2

[13] Rewon Child. Very deep VAEs generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*, 2021. 13

[14] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. 12, 16

[15] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. RandAugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, volume 33, pages 18613–18624. Curran Associates, Inc., 2020. 11

[16] Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*, 2019. 19

[17] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. CINIC-10 is not ImageNet or CIFAR-10. *arXiv preprint arXiv:1810.03505*, 2018. 4

[18] James Diffenderfer, Brian Bartoldson, Shreya Chaganti, Jize Zhang, and Bhavya Kailkhura. A winning hand: Compressing deep networks can improve out-of-distribution robustness. *Advances in Neural Information Processing Systems*, 34, 2021. 12

[19] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation, 2014. 2

[20] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP, 2016. 13

[21] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning, 2017. 2

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 4, 10

[23] Ethan Fetaya, Joern-Henrik Jacobsen, Will Grathwohl, and Richard Zemel. Understanding the limitations of conditional generative models. In *International Conference on Learning Representations*, 2020. 13

[24] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 1

[25] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 12

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 3, 14

[27] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 2, 12, 16

[28] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. *International Conference on Learning Representations*, 2019. 13

[29] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020. 1, 12, 16

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 10, 15

[31] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8340–8349, October 2021. 4

[32] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 4, 16, 17

[33] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations*, 2019. 4, 12, 15

[34] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. 4, 12, 15, 16

[35] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019. 3, 12

[36] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in Neural Information Processing Systems*, 32, 2019. 12

[37] Dan Hendrycks*, Norman Mu*, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*, 2020. 3, 4, 12, 16

[38] Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. PixMix: Dreamlike pictures comprehensively improve safety measures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16783–16792, 2022. 4, 16

[39] 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. In *International Conference on Learning Representations*, 2016. 13

[40] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. 13

[41] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[42] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 10, 16, 17

[43] Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. In *International Conference on Learning Representations*, 2019. 2, 13

[44] Jörn-Henrik Jacobsen, Arnold W.M. Smeulders, and Edouard Oyallon. i-RevNet: Deep invertible networks. In *International Conference on Learning Representations*, 2018. 13

[45] Ayush Jaiswal, Rob Brekelmans, Daniel Moyer, Greg Ver Steeg, Wael AbdAlmageed, and Premkumar Natarajan. Discovery and separation of features for invariant representation learning, 2019. 13

[46] Ayush Jaiswal, Rex Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Unsupervised adversarial invariance. *Advances in Neural Information Processing Systems*, 31, 2018. 2, 13

[47] Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. *Advances in Neural Information Processing Systems*, 33:10558–10570, 2020. 16

[48] Jongheon Jeong and Jinwoo Shin. Training GANs with stronger augmentations via contrastive discriminator. In *International Conference on Learning Representations*, 2021. 19

[49] Daniel Kang, Yi Sun, Tom Brown, Dan Hendrycks, and Jacob Steinhardt. Transfer of adversarial robustness between perturbation types, 2019. 12

[50] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12104–12114. Curran Associates, Inc., 2020. 11, 19

[51] 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*, pages 4401–4410, 2019. 10

[52] 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*, pages 8110–8119, 2020. 11, 19

[53] 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. 16

[54] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 13, 17

[55] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. 10

[56] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 13

[57] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016. 13

[58] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014. 2, 13, 14

[59] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020. 13

[60] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990. 2

[61] Artemy Kolchinsky, Brendan D Tracey, and David H Wolpert. Nonlinear information bottleneck. *Entropy*, 21(12):1181, 2019. 17

[62] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009. 1, 15, 19

[63] Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in GANs. In *International Conference on Machine Learning*, pages 3581–3590. PMLR, 2019. 10

[64] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. 17

[65] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018. 12

[66] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 1, 4, 12

[67] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3763–3772. PMLR, 09–15 Jun 2019. 12

[68] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. 4

[69] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized GAN training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations*, 2021. 10, 11, 19

[70] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020. 4

[71] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 19

[72] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts, 2016. 10, 11

[73] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 11

[74] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder, 2015. 13

[75] Shangyun Lu, Bradley Nott, Aaron Olson, Alberto Todeschini, Hossein Vahabi, Yair Carmon, and Ludwig Schmidt. Harder or different? a closer look at distribution shift in dataset reproduction. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2020. 4

[76] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 3, 12, 16

[77] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 1, 13, 19

[78] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018. 11

[79] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pages 7721–7735. PMLR, 2021. 4

[80] Norman Mu and Justin Gilmer. MNIST-C: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019. 17

[81] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019. 13

[82] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 12

[83] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[84] Ziqi Pan, Li Niu, Jianfu Zhang, and Liqing Zhang. Disentangled information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9285–9293, 2021. 2, 13, 17, 18

[85] 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. 19

[86] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv preprint arXiv:1806.00451*, 2018. 4

[87] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 13

[88] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1, 4, 15

[89] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected GANs converge faster. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 11, 19

[90] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on MNIST. In *International Conference on Learning Representations*, 2019. 12

[91] Joan Serra, David Alvarez, Vicenc Gomez, Olga Slizovskaia, Jose F. Nunez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations*, 2020. 13

[92] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 13

[93] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. 2, 12, 16

[94] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. CSI: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems*, 2020. 12, 15, 16

[95] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020. 4

[96] R Thobaben, M Skoglund, et al. The convex information bottleneck lagrangian. *Entropy*, 22(1), 2020. 17

[97] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999. 1, 13

[98] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop*, pages 1–5, 2015. 1, 2

[99] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 4, 10

[100] Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. *Advances in Neural Information Processing Systems*, 32, 2019. 12

[101] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020. 13, 19

[102] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery. 13

[103] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 12

[104] Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20685–20696. Curran Associates, Inc., 2020. 12, 13

[105] Jingkang Yang, Kaiyang Zhou, and Ziwei Liu. Full-spectrum out-of-distribution detection. *arXiv preprint arXiv:2204.05306*, 2022. 4

[106] Xiulong Yang and Shihao Ji. JEM++: Improved techniques for training JEM. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6494–6503, October 2021. 12

[107] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 11

[108] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 11

[109] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled

trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019. 12

[110] Zijun Zhang, Ruixiang Zhang, Zongpeng Li, Yoshua Bengio, and Liam Paull. Perceptual generative autoencoders. In *International Conference on Machine Learning*, pages 11298–11306. PMLR, 2020. 19

[111] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient GAN training. *arXiv preprint arXiv:2006.10738*, 2020. 11, 19

A. Training procedure of NIBAE

Algorithm 1 Nuisance-extended information bottleneck autoencoder (NIBAE)

Require: encoder f, decoder g, discriminators d, prior $p_0(\mathbf{z})$, $\alpha, \beta, \tau > 0$.

for # training iterations **do**
  Sample $(x_i, y_i)_{i=1}^m \sim p_d(\mathbf{x}, \mathbf{y})$
  $\mathbf{z}^{(i)}, \mathbf{z}_n^{(i)} \leftarrow f(x_i)$, and sample $z^{(i)}, z_n^{(i)} \sim \mathbf{z}^{(i)}, \mathbf{z}_n^{(i)}$
  $\hat{x}_i \leftarrow g(z^{(i)}, z_n^{(i)})$
  // UPDATE DISCRIMINATORS
  $d_{\texttt{sim}}^{(i)} \leftarrow \mathrm{sim}(d_{\mathbf{x}}(\Pi_E(x_i)), d_{\mathbf{x}}(\Pi_E(\hat{x}_i)))/\tau$
  $L_{\texttt{sim}} \leftarrow \frac{1}{m}\sum_i \log(1 - \mathrm{sigmoid}(d_{\texttt{sim}}^{(i)}))$
  $L_{\texttt{ind}} \leftarrow \mathbb{E}_{\mathbf{z},\mathbf{z}_n \sim \mathcal{N}(0,I)}[\log d_{\mathbf{z}}(\mathbf{z}, \mathbf{z}_n)] + \frac{1}{m}\sum_i \log(1 - d_{\mathbf{z}}(z, z_n))$
  $L_{\texttt{nuis}}^D \leftarrow \frac{1}{m}\sum_i \mathbb{CE}(q_n(\mathbf{y}|z_n^{(i)}), y_i)$
  $L_D \leftarrow L_{\texttt{nuis}}^D - L_{\texttt{ind}} - L_{\texttt{sim}}$
  $d_{\mathbf{x}}, d_{\mathbf{z}}, q_n \leftarrow$ Update $d_{\mathbf{x}}, d_{\mathbf{z}}, q_n$ to minimize L_D
  // UPDATE ENCODER AND DECODER
  $L_{\texttt{VIB}}^\beta \leftarrow \frac{1}{m}\sum_i [-\log q(y_i|z_i) + \beta \mathrm{KL}(p(\mathbf{z}|x_i)\|p_0(\mathbf{z}))]$
  $L_{\texttt{recon}} \leftarrow \frac{1}{m}\sum_i \frac{1}{2}\|x_i - \hat{x}_i\|_2^2$
  $L_{\texttt{nuis}} \leftarrow \frac{1}{m}\sum_i \mathbb{CE}(q_n^*(\mathbf{y}|z_n^{(i)}), \frac{1}{|\mathcal{Y}|})$
  $L_{\texttt{NIBAE}} \leftarrow L_{\texttt{VIB}}^\beta + \alpha L_{\texttt{recon}} + L_{\texttt{nuis}} + L_{\texttt{ind}} + L_{\texttt{sim}}$
  $f, g, \Pi_f \leftarrow$ Update f, g, Π_f to minimize $L_{\texttt{NIBAE}}$
end for

B. Experimental details

B.1. Architectures

Recall that our proposed NIBAE architecture consists of (a) an encoder f, (b) a decoder g, and (c) MLP-based discriminators $d_{\mathbf{y}}$, $d_{\mathbf{z}}$, and an MLP for feature statistic projection Π_f. For the encoder architecture, we mainly consider ResNet-18 [30] and ViT-S [22, 99] in our experiments. When ResNet-18 is used, we consider the generator architecture of FastGAN [69] as the decoder, but with a modification on normalization layers: specifically, we replace the standard batch normalization [42] layers in the architecture with adaptive instance normalization (AdaIN) [51] so that the affine parameters can be modulated by $\mathbf{z}$ and $\mathbf{z}_n$ as well as the decoder input: we observe a consistent gain in FID from this modification. In cases when ViT-S is used as the encoder, on the other hand, we use the same transformer architecture as the decoder model where it is preceded by linear layers that maps both $\mathbf{z}$ and $\mathbf{z}_n$ into the space of patch embedding. We assume the patch size of ViT as 4, *i.e.*, we denote it as ViT-S/4, so that the output from the model contains 8×8 patch embeddings in case of CIFAR-10 similarly to the ResNet-18 architecture. To model $\mathbf{z}$ and $\mathbf{z}_n$ in the ViT architecture, we simply split the output patch embedding into two separate embeddings (of reduced embedding dimensions): one of these embeddings is average-pooled to define $\mathbf{z}$, and the remaining one is used as the nuisance $\mathbf{z}_n$. We set 128 as the nuisance dimension $\mathbf{z}_n$, and use hidden layer of size 1,024 for MLP-based discriminators, *e.g.*, $d_{\mathbf{y}}$, $d_{\mathbf{z}}$, and MLPs for projection Π_f.

B.2. Training and hyperparameters

Unless otherwise noted, we train each model for 200K updates. For training NIBAE models, we use $\alpha = 0.01, \beta = 0.0001$, and $\tau = 0.2$ unless otherwise noted. We use different training configurations depending on the encoder architecture, *i.e.*, whether is it ResNet-18 or ViT-S/4: (a) For ResNet-based models, we train the encoder part (f) via stochastic gradient descent (SGD) with batch size of 64 using Nesterov momentum of weight 0.9 without dampening. We set a weight decay of 10^{-4}, and use the cosine learning rate scheduling [72] from the initial learning rate of 0.1. For the remainder parts of our NIBAE architecture, *e.g.*, the decoder g and discriminator MLPs, on the other hand, we follow the training practices of GAN instead: specifically, we use Adam [55] with $(\alpha, \beta_1, \beta_2) = (0.0002, 0.5, 0.999)$, following the hyperparameter practices explored by [63]. (b) For ViT-based models, on the other hand, we train both (transformer-based) encoder and decoder models

via AdamW [73] with a weight decay of 10^{-4}, using batch size 128 and $(\alpha, \beta_1, \beta_2) = (0.0002, 0.9, 0.999)$ with the cosine learning rate scheduling [72] and 2000 steps of a linear warm-up phase in learning rate. Overall, we observe that a stable training of ViT on CIFAR-10 requires much stronger regularization compared to ResNets, otherwise they often significantly suffer from overfitting. In this respect, we apply various regularization practices those are now widely used for ViTs on ImageNet-1K, namely mixup [108], CutMix [107], and RandAugment [15], following those established in [8]: which could lead a stable ViT training on CIFAR-10 achieving similar performance to ResNet in terms of (clean) test accuracy.

B.3. StyleGAN2 and FastGAN

For the experiments reported in Table 7 of the main text, we adopt StyleGAN2 [52] and FastGAN [69] architectures to verify the effectiveness of our proposed FSD. For the StyleGAN2-based models, we follow the training details of DiffAug [111] and ADA [50] in their CIFAR experiments: specifically, we use Adam with $(\alpha, \beta_1, \beta_2) = (0.002, 0.0, 0.99)$ for optimization with batch size of 64. We use non-saturating loss for training, and use R_1 regularization [78] with $\gamma = 0.01$. We do not use, however, the path length regularization and the lazy regularization [52] in training. We take exponential moving average on the generator weights with half-life of 500K samples. We stop training after 800K generator updates, which is about the half of those conducted for the ADA baseline [50]. For the FastGAN baseline, on the other hand, we run the official implementation of FastGAN[8] [69] on CIFAR-10 for the length of 6.4M samples with batch size 16. For the "Projected GAN" baseline, we adapt the official implementation[9] [89] onto the ImageNet pre-trained ResNet-18, and trained for 6.4M samples with batch size 64. Our results ("FSD") follows the same training details, but with a difference in its discriminator architecture.

B.4. Computing infrastructure

Unless otherwise noted, we use a single NVIDIA Geforce RTX-2080Ti GPU to execute each of the experiments. For experiments based on StyleGAN2 architecture (Table 7), we use two NVIDIA Geforce RTX-2080Ti GPUs per run.

C. Ablation study

Figure 4. Reconstructions under random nuisance $\mathbf{z}_n$. The leftmost per row shows the original reconstruction.

Table 3. Comparison of the test error rate (Err.; %), corruption error (C-Err.; %) and FID on CIFAR-10 across ablations.

β	$L_{\texttt{recon}}$	$L_{\texttt{sim}}$	$L_{\texttt{nuis}}$	$L_{\texttt{ind}}$	Err.	C-Err.	FID
1e-4	✓	✓	✓	✓	7.07	23.3	33.3
1e-3	✓	✓	✓	✓	7.32	24.5	31.0
1e-2	✓	✓	✓	✓	7.38	26.3	30.8
1e-4	✗	✓	✓	✓	8.29	29.2	33.8
1e-4	✓	✗	✓	✓	7.95	28.6	83.1
1e-4	✓	✓	✗	✓	8.01	24.1	29.2
1e-4	✓	✓	✓	✗	7.31	22.4	78.3

We further perform an ablation study on CIFAR-10 for a detailed analysis of the proposed NIBAE:

Effect of β. As also introduced in the original IB objective, $\beta \geq 0$ plays the key role in NIBAE training as it controls the information balance between the semantic $\mathbf{z}$ and the nuisance $\mathbf{z}_n$. Here, Figure 4 examine how using different value of β affect the actual representations, by comparing the reconstructed samples for a fixed input while randomizing the nuisance $\mathbf{z}_n$. Indeed, we observe a clear trend from this comparison demonstrating the effect of β: having larger β makes the model to push more "semantic" information into $\mathbf{z}_n$ regarding it as the nuisance. Without information bottleneck, *i.e.*, in case when $\beta = 0.0$, we qualitatively observe that the network rather encodes most information in $\mathbf{z}$, due to the minimax loss applied to the nuisance $\mathbf{z}_n$. Quantitatively, this behavior is further evidenced in Table 3 as an increase in the corruption errors when using larger β.

Reconstruction loss. The reconstruction loss $L_{\texttt{recon}}$ is one of essential part to make NIBAE work as a "nuisance modeling": in Table 3, we provide an ablation when this loss is omitted, showing a significant degradation in the final accuracy, and more crucially in the corruption error. This confirms the necessity of reconstruction loss to obtain a robust representation in NIBAE. Nevertheless, due to the adversarial similarity loss $L_{\texttt{sim}}$ that can also work (while not perfectly) as a reconstruction loss, one can still observe that the FID of the model can be moderately preserved.

[8] https://github.com/odegeasslbc/FastGAN-pytorch
[9] https://github.com/autonomousvision/projected_gan

Adversarial similarity. When the $L_{\texttt{sim}}$ is omitted, on the other hand, we instead observe a significant degradation in FID rather than accuracy, showing the effectiveness of our proposed adversarial similarity based guidance to improve decoder performance while affecting less to the accuracy compared to the case when $L_{\texttt{recon}}$ is ablated. It is quite remarkable that there is still a degradation in both clean and corruption accuracies compared to the case when $L_{\texttt{sim}}$ is jointly minimized: we observe that in this scenario of missing $L_{\texttt{sim}}$, the overall reconstruction loss $L_{\texttt{recon}}$ is often also less optimized, which could eventually affect the quality of $\mathbf{z}$.

Nuisance loss. From the ablation of $L_{\texttt{nuis}}$ given in Table 3, we observe not only a considerable degradation in clean accuracy but also in its corruption robustness. This shows that strictly forcing the nuisance-ness to $\mathbf{z}_n$ (against $\mathbf{y}$) indeed helps $\mathbf{z}$ to learn a more robust representation, possibly from encouraging $\mathbf{z}$ to extract more diverse class-related information in a faithful manner by keeping the remainder information in $\mathbf{z}_n$ sufficient to infer $\mathbf{x}$.

Independence loss. The independence loss $L_{\texttt{ind}}$ in our current design, which essentially performs a GAN training toward $p(\mathbf{z}, \mathbf{z}_n) \sim \mathcal{N}(0, I)$, not only forces $\mathbf{z} \perp \mathbf{z}_n$ but also leads $\mathbf{z}$ and $\mathbf{z}_n$ to have a tractable marginal distribution: so that one could efficiently perform a sampling from the learned decoder. In a practical aspect, therefore, omitting $L_{\texttt{ind}}$ in NIBAE can directly harm its generation quality as given in Table 3. Nevertheless, it is still remarkable that the ablation could rather improve the corruption error: this suggests that our current design of forcing the full Gaussian may be restrictive. An alternative design for the future work could assume a weaker condition for $\mathbf{z}$ and $\mathbf{z}_n$, instead with a more sophisticated sampling to obtain a valid generative model from NIBAE.

D. Proof of Lemma 1

Lemma 1. *Let $\mathbf{x} \in \mathcal{X}$, and $\mathbf{y} \in \mathcal{Y}$ be random variables, $\hat{\mathbf{x}}$ be a noisy observation of $\mathbf{x}$ with $I(\mathbf{x};\mathbf{y}) = I(\hat{\mathbf{x}};\mathbf{y})$. Given that a representation $[\hat{\mathbf{z}}, \hat{\mathbf{z}}_n] := f(\hat{\mathbf{x}})$ of $\hat{\mathbf{x}}$ satisfies (a) $H(\hat{\mathbf{x}}|\hat{\mathbf{z}}, \hat{\mathbf{z}}_n) = 0$, (b) $I(\hat{\mathbf{z}}_n;\mathbf{y}) = 0$, and (c) $\hat{\mathbf{z}} \perp \hat{\mathbf{z}}_n$, it holds $I(\hat{\mathbf{z}};\mathbf{y}) = I(\mathbf{x};\mathbf{y})$.*

Proof. Given that f is invertible for the random variable $\hat{\mathbf{x}}$, the statement follows from the chain rule of mutual information and that of conditional mutual information, as well as by applying (b) and (c):

$$I(\mathbf{x};\mathbf{y}) = I(\hat{\mathbf{x}};\mathbf{y}) = I(\mathbf{y};\hat{\mathbf{z}},\hat{\mathbf{z}}_n) = I(\mathbf{y};\hat{\mathbf{z}}_n) + I(\mathbf{y};\hat{\mathbf{z}}|\hat{\mathbf{z}}_n) \tag{11}$$

$$= I(\mathbf{y};\hat{\mathbf{z}}) + H(\hat{\mathbf{z}}_n|\mathbf{y}) + H(\hat{\mathbf{z}}_n|\hat{\mathbf{z}}) - H(\hat{\mathbf{z}}_n|\mathbf{y},\hat{\mathbf{z}}) - H(\hat{\mathbf{z}}_n) \tag{12}$$

$$= I(\mathbf{y};\hat{\mathbf{z}}) = I(\hat{\mathbf{z}};\mathbf{y}). \tag{13}$$

□

E. Additional background

E.1. Detailed survey on related work

Out-of-distribution robustness. Since the seminal works [2, 82, 93] revealing the fragility of neural networks for out-of-distribution inputs, there have been significant attempts on identifying and improving various notions of robustness: *e.g.*, detecting novel inputs [34–36, 65, 66, 94, 104], robustness against corruptions [18, 25, 33, 37, 103], and adversarial noise [5, 9, 14, 27, 76, 109], to name a few. Due to its fundamental challenges in making neural network to extrapolate, however, most of the advances in the robustness literature has been made under assuming priors closely related to the individual problems: *e.g.*, *Outlier Exposure* [35] and *AugMix* [37] assume an external dataset or a pipeline of data augmentations to improve the performances in novelty detection and corruption robustness, respectively; *Tent* [103] leverages extra information available from a batch of samples in test-time to adapt a given neural network; [49, 100] observe that neural networks robust to a certain type of adversarial attack (*e.g.*, an ℓ_∞-constrained adversary) do not necessarily robust to other types of adversary (*e.g.*, an ℓ_1-constrained adversary), *i.e.*, adversarial robustness hardly generalizes from the adversary assumed *a priori* for training. In this work, we aim to improve multiple notions of robustness without assuming such priors, through a new training scheme that extends the standard information bottleneck principle under noisy observations in test-time.

Hybrid generative-discriminative modeling. Our proposed method can be also viewed as a new approach of improving the robustness of discriminative models by incorporating a generative model, in the context that has been explored in recent works [29, 66, 90, 106]: for example, [66, 67] have shown that assuming a simple Gaussian mixture model on the deep discriminative representations can improve novelty detection and robustness to noisy labels, respectively; [90] develop an empirical defense against adversarial examples via generative classifiers; A line of research on *Joint Energy-based Models* (JEM) [29, 106] assumes the entire discriminative model as a joint generative model by interpreting the logits of $p(\mathbf{y}|\mathbf{x})$ as unnormalized log-densities of $p(\mathbf{x}|\mathbf{y})$, and shows that modeling $p(\mathbf{x}|\mathbf{y})$ as well as $p(\mathbf{y}|\mathbf{x})$ can improve

out-of-distribution generalization of the classifier. Nevertheless, it is still an unexplored and open question that how to "better" incorporate generative representation into discriminative models: in case of novelty detection, for example, several recent works [81, 87, 91, 104] observe that existing likelihood-based generative models are not accurate enough to detect out-of-distribution datasets, suggesting that relying solely on (likelihood-based) deep generative representation may not enough for robust classification [23]. In case of JEM, on the other hand, it has been shown that directly assuming a joint generative-discriminative representation often makes a significant training instability. In this work, we propose to introduce an autoencoder-based model to avoid the training instability, and consider a design that the *nuisance* can succinctly supplement the given discriminative representation to be generative.

Invertible representations and nuisance modeling. The idea of incorporating nuisances can be also considered in the context of *invertible* modeling, or as known as *flow-based models* [6, 11, 20, 28, 44, 56],[10] which maps a given input $\mathbf{x}$ into a representation $\mathbf{z}$ of the same dimension so that one can construct an inverse of $\mathbf{z}$ to $\mathbf{x}$: here, the nuisance can be naturally defined as the remainder information of $\mathbf{z}$ for a given subspace of interest, *e.g.*, to model $\mathbf{y}$. For example, [43] adopt a fully-invertible variant of i-RevNet [44] to analyze *excessive invariance* in neural networks, *i.e.*, the existence of pairs of completely different samples with the same representation in a neural network, and proposes to maximize the cross-entropy for the nuisances in a similar manner to our proposed minimax-based nuisance loss ((6) in the main text); [4], on the other hand, leverages invertible neural network to model a Gaussian mixture based generative classifier in the representation space, so that nuisance information can be preserved until its representation. Compared to such approaches relying on invertible neural networks, our autoencoder-based nuisance modeling does not guarantee the "full" invertibility for arbitrary inputs: instead, it only focuses on inverting the data manifold given, and this enables (a) a much flexible encoder design in practice, *i.e.*, other than flow-based designs, and (b) a more scalable generative modeling of nuisance representation $\mathbf{z}_n$ while forcing its *independence* to the semantic space $\mathbf{z}$. This is due to that it works on a compact space rather than those proportional to the input dimension, which is an important benefit of our modeling in terms of the scalability of nuisance-aware training, *e.g.*, beyond an MNIST-scale as done in [43]. More closer related works [45, 46, 84] in this respect instead introduce a separate encoder for nuisance factors, where the nuisanceness is induced by the independence to $\mathbf{z}$: *e.g.*, DisenIB [84] applies FactorVAE [54] between semantic and nuisance embeddings to force their independence.[11] Yet, similarly to the invertible approach, the literature has been questioned on that the idea can be scaled-up beyond, *e.g.*, MNIST, and our work does explore and establish a practical design that is applicable for recent architectures and datasets addressing modern security metrics, *e.g.*, corruption robustness. On the technical side, for example, we find that the "nuisanceness to $\mathbf{y}$" is more important for $\mathbf{z}_n$ than the "independence with $\mathbf{z}$" (as usually done in the previous works [45, 46, 84]) to induce a robust representation, as verified in our ablation study in Appendix C, which can be a useful practice for the future research concerning robust representation learning.

Autoencoder-based generative models. There have been steady advances in generative modeling based on autoencoder architectures, especially since the development in *variational autoencoders* (VAEs) [58]: due to its ability of estimating data likelihoods, and its flexibility to implement various statistical assumptions [54, 57, 74]. With the advances in its training objectives [39, 77, 102] as well as the architectural improvements [13, 101], VAE-based models are currently considered as one of state-of-the-art approaches in likelihood based generative modeling: *e.g.*, a state-of-the-art *diffusion models* [40, 92] is built upon the denoising autoencoders under Gaussian perturbations, and recently-proposed *hierarchical VAEs* [13, 101] have shown that VAEs can benefit from scaling up its architectures into deeper encoder networks. In perspectives of viewing our method as a *generative modeling*, NIBAE is based on *adversarial autoencoders* [77] that replaces the KL-divergence based regularization in standard VAEs with a GAN-based adversarial loss, with a novel encoder architecture that is based on the *internal feature statistics* of discriminative models: so that the model can better encode lower-level features without changing the backbone architecture. We observe that this design enables autoencoder-based modeling even from a large, pre-trained discriminative models, and this "projection" of internal features can significantly benefit the generation quality, as well as for generative adversarial networks (GANs) as observed in Table 7 in the main text.

E.2. Technical background

Variational information bottleneck. Although the information bottleneck (IB) principle given in (1) [97] suggests a useful definition on what we mean by a "good" representation, computing mutual information of two random variables is generally hard and this makes the IB objective infeasible in practice. To overcome this, *variational information bottleneck* (VIB) [1, 10] applies variational inference to obtain a lower bound on the IB objective (1). Specifically, it approximates: (a) $p(\mathbf{y}|\mathbf{z})$ by a (parametrized) "decoder" neural network $q(\mathbf{y}|\mathbf{z})$, and (b) $p(\mathbf{z})$ by an "easier" distribution $r(\mathbf{z})$, *e.g.*, isotropic Gaussian

[10] A more complete survey on flow-based models can be found in [59].

[11] We provide a more direct empirical comparison with DisenIB [84] to our proposed method in Appendix G.4.

$\mathcal{N}(\mathbf{z}|0, I)$. Having such (variational) approximations in computing (1) as well as the Markov chain property $\mathbf{y} - \mathbf{x} - \mathbf{z}$ of neural networks, one yields the following lower bound on the IB objective (1):

$$I(\mathbf{z}; \mathbf{y}) - \beta I(\mathbf{z}, \mathbf{x}) \geq \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\int dz \left(p(z|\mathbf{x}) \log q(\mathbf{y}|z) - \beta p(z|\mathbf{x}) \log \frac{p(z|\mathbf{x})}{r(z)} \right) \right]. \tag{14}$$

This bound can now be approximated with the empirical distribution $p(\mathbf{x}, \mathbf{y}) \approx \frac{1}{n} \sum_i \delta_{x_i}(\mathbf{x}) \delta_{y_i}(\mathbf{y})$ from data. By further assuming a Gaussian encoder $p(\mathbf{z}|\mathbf{x}) := \mathcal{N}(\mathbf{z}|f^{\mu}(\mathbf{x}), f^{\sigma}(\mathbf{x}))$ as defined in (2) and applying the reprarametrization trick [58], we get the following VIB objective:

$$L^{\beta}_{\texttt{VIB}} := \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{\epsilon}}[-\log q(y_i|f(x_i, \boldsymbol{\epsilon}))] + \beta \, \mathrm{KL} \left(p(\mathbf{z}|x_i) \| r(\mathbf{z}) \right). \tag{15}$$

Generative adversarial networks. *Generative adversarial network* (GAN) [26] considers the problem of learning a generative model p_g from given data $\{x_i\}_{i=1}^n$, where $x_i \sim p_d(\mathbf{x})$ and $\mathbf{x} \in \mathcal{X}$. Specifically, GAN consists of two neural networks: (a) a *generator* network $G : \mathcal{Z} \to \mathcal{X}$ that maps a latent variable $z \sim p(\mathbf{z})$ into $\mathcal{X}$, where $p(\mathbf{z})$ is a specific prior distribution, and (b) a *discriminator* network $D : \mathcal{X} \to [0, 1]$ that discriminates samples from p_d and those from the implicit distribution p_g derived from $G(\mathbf{z})$. The primitive form of training G and D is the following:

$$\min_G \max_D V(G, D) := \mathbb{E}_{\mathbf{x}}[\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}}[\log(1 - D(G(\mathbf{z})))]. \tag{16}$$

For a fixed G, the inner maximization objective (16) with respect to D leads to the following *optimal discriminator* D^*_G, and consequently the outer minimization objective with respect to G becomes to minimize the *Jensen-Shannon divergence* between p_d and p_g, namely $D^*_G := \frac{p_d}{p_d + p_g}$.

F. Architectures for nuisance modeling

In principle, our framework is generally compatible with existing any deep network architectures: *e.g.*, say an encoder $E : \mathcal{X} \to \mathcal{Z}$ and decoder $G : \mathcal{Z} \to \mathcal{X}$, respectively. In order to apply VIB, we assume that the encoder has two output heads of dimension $2K$, where K denotes the size of latent representation $\mathbf{z}$: here, each output head models the Gaussian random variable by reparametrization, *i.e.*, by modeling (μ, σ) as the encoder output for both $\mathbf{z} \in \mathbb{R}^K$ and $\mathbf{z}_n \in \mathbb{R}^{K_n}$.

Although it is possible that the encoder E models representations $\mathbf{z}$ and $\mathbf{z}_n$ by simply taking outputs from a deep feed-forward representations following conventions, we observe that modeling nuisances $\mathbf{z}_n$ as well as $\mathbf{z}$, which is essentially "generative", in standard discriminative architectures can incur a bottleneck in performance compared to modeling with the nuisance $\mathbf{z}_n$: the nuisance information often requires to model the fine details in a given inputs, which is available in early layers of E, but may not in the later layers for classification. Motivated by the following observations we made for GANs, therefore, we propose to encode nuisance $\mathbf{z}_n$ as well as the ("semantic") representation $\mathbf{z}$ from the *collection* of interal features statistics, rather than by a mapping from the last layer of E.

Motivation: Feature statistics discriminator (FSD) for GANs. Designing a stable discriminator D is a key aspect for a successful training of GANs. The usual practice in the GAN literature is to have a separate discriminator network with a comparable size to G, but this can be a significant computational (and memory) overhead in the framework. Instead of having a separate discriminator network, we observe that the *internal feature statistics* of the encoder E can be a surprisingly effective representation to define a simple yet efficient D: Concretely, for a given encoder E and an input $\mathbf{x}$, we consider L intermediate feature maps of $\mathbf{x}$, namely $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(L)}$ from $E(\mathbf{x})$, and define the *projection* of $\mathbf{x}$ as the following:

$$\Pi_E(\mathbf{x}) := \begin{bmatrix} \mathbf{m}^{(1)} & \mathbf{m}^{(2)} & \cdots & \mathbf{m}^{(L)} \\ \mathbf{s}^{(1)} & \mathbf{s}^{(2)} & \cdots & \mathbf{s}^{(L)} \end{bmatrix}, \tag{17}$$

where $\mathbf{m}^{(l)}$ and $\mathbf{s}^{(l)}$ are the first- and second moment of channel-wise feature maps in $\mathbf{x}^{(l)}$, assuming that $\mathbf{x}^{(l)} \in \mathbb{R}^{HWC}$ follows the format of convolutional feature maps:

$$\mathbf{m}^{(l)}_c := \frac{1}{HW} \sum_{h,w} \mathbf{x}^{(l)}_{h,w,c}, \text{ and } \mathbf{s}^{(l)}_c := \frac{1}{HW} \sum_{h,w} (\mathbf{x}^{(l)}_{h,w,c} - \mathbf{m}^{(l)}_c)^2. \tag{18}$$

The *features statistics discriminator* (FSD) we consider here is then simply a 3-layer multi-layer perceptron (MLP) applied on $\Pi_E(\mathbf{x})$. Table 7 compares and confirms that this simplest design of GAN discriminator can significantly accelerate GANs

when applied to pre-trained discriminative encoder architectures (here we use ResNet-18 or ResNet-50 [30] pre-trained on ImageNet [88]).

Returning to our encoder design, motivated by that the features statistics based projection Π_E can better encode generative representation in discriminative models, we apply the same idea to model the encoder representations $\mathbf{z}$ and $\mathbf{z}_n$: specifically, we model $\mathbf{z}$ and $\mathbf{z}_n$ by simply applying MLPs to the feature statistics projection $\Pi_E(\mathbf{x})$ (17): we indeed observe this enables faster and stable training, and it even allows auto-encoder based training from a pre-trained discriminative model, in a similar manner done in Table 7, as examined in Table 8 (see Appendix I for more details).

Adversarial similarity based guidance. In addition to the objectives considered in Section 2 that are essential for NIBAE, particularly for ConvNet-based models, *e.g.*, , ResNet-18 as we consider in the experiments, we found that it is useful to further leverage our proposed *feature statistics* based encoder architecture (see Section F) to provide the decoder g an extra guidance in minimizing the (pixel-level) reconstruction loss (5): specifically, we propose to additionally place a discriminator network, say $d_{\mathbf{x}} : \mathbb{R}^{|\Pi_f|} \to \mathbb{R}^e$, that computes similarity between $\Pi_f(\mathbf{x})$ and $\Pi_f(G(\mathbf{z}, \mathbf{z}_n))$ and performs adversarial training on it:

$$L_{\texttt{sim}} := \log(1 - \text{sigmoid}(s(d^*_{\mathbf{x}})), \text{where } s(d_{\mathbf{x}}) := \text{sim}(d_{\mathbf{x}}(\Pi_f(\mathbf{x})), d_{\mathbf{x}}(\Pi_f(g(\mathbf{z}, \mathbf{z}_n))))/\tau. \tag{19}$$

Here, $d^*_{\mathbf{x}} := \max_{d_{\mathbf{x}}} L_{\texttt{sim}}(d_{\mathbf{x}})$, $\text{sim}(\mathbf{x}, \mathbf{y}) := \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$ denotes the cosine similarity, and τ is the temperature hyperparameter. We use $\tau = 0.2$ throughout our experiments.

G. Additional experimental results

G.1. Out-of-distribution detection

Score functions. A typical practice to address out-of-distribution task is to assign a *score function* for each input based on the model, *e.g.*, the maximum confidence score [34] as commonly used for supervised models, to threshold out samples as out-of-distribution when the score is low. To define a score function for our NIBAE models, we first observe that the log-likelihood score of the nuisance representation $\mathbf{z}_n$, which is a unique information for NIBAE, can be a strong score function especially for detecting novelties those are semantically far from in-distribution, *i.e.*,

$$\log \mathcal{N}(\mathbf{z}_n; 0, I) = -\tfrac{1}{2}\|\mathbf{z}_n\|^2, \tag{20}$$

as we assume that $\mathbf{z}$ follows isotropic Gaussian $\mathcal{N}(0, I)$. For detecting so-called "harder" novelties, we propose to use the log-likelihood score of $\mathbf{y}$ under a *symmetric Dirichlet distribution* of parameter $\alpha > 0$, namely $\text{Dir}_\alpha(\mathbf{y}) \in \Delta^{|\mathcal{Y}|-1}$, rather than simply using $\max_y p(y|x)$: *i.e.*,

$$\log \text{Dir}_\alpha(\mathbf{y}) = (\alpha - 1) \sum_i \log y_i. \tag{21}$$

Note that the distribution gets closer to the symmetric (discrete) one-hot distribution as $\alpha \to 0$, which makes sense for most classification tasks, and here we simply use $\alpha = 0.05$ throughout experiments.[12]

In Table 4, we compare AUROC on detecting standard OOD benchmarks from models trained on CIFAR-10: overall, we confirm that the score function combining the information of $\mathbf{z}_n$ and $\mathbf{y}$ of NIBAE improve novelty detection in a complementary manner, showing the effectiveness of modeling nuisance. For example, the combined score achieves near-perfect AUROCs for detecting SVHN, LSUN and ImageNet datasets. It is also remarkable that NIBAE could also outperform a strong baseline of (a supervised version of) CSI [94], which further utilizes an "OOD-like" augmentations in their representation learning.

G.2. Robustness against common corruptions

Table 5 report the corruption robustness on CIFAR-10/100-C [33], a corrupted version of CIFAR-10/100 [62] equipped with 15 natural corruptions under 5 different severity levels. Overall, we observe that NIBAE notably and consistently improves the robustness at common corruptions than classifiers trained with cross-entropy or combined with naïve VIB objectives, in both architectures of ResNet-18 and ViT-S/4 tested. Interestingly, we observe that the impact of NIBAE in the clean error can be different depending on the encoder architecture: with the ViT-S, NIBAE could even further improve the clean errors compared to both Cross-entropy and VIB. This is possibly due to that the representation induced via NIBAE can be extracted better with non-local (attention-based) operations.

[12] In practice, we observe that other choices in a moderate range of α near 0 do not much affect performance.

Table 4. AUROC values of various OOD detection methods trained on the CIFAR-10 dataset with five OOD datasets: SVHN, LSUN, ImageNet, CIFAR-100, and CelebA. Bolds indicate the best results.

Method	Score	SVHN	LSUN	ImageNet	CIFAR-100	CelebA
JEM [29]	$\log p(x)$	0.67	-	-	0.67	0.75
JEM [29]	$\max_y p(y\|x)$ [34]	0.89	-	-	0.87	0.79
SupCon [53]	$\max_y p(y\|x)$ [34]	0.97	0.93	0.91	0.89	-
CSI [94]	$\max_y p(y\|x)$ [34]	**0.98**	0.98	0.98	**0.92**	-
Cross-entropy	$\max_y p(y\|x)$ [34]	0.94	0.94	0.92	0.86	0.64
Cross-entropy	$\log \mathrm{Dir}_{0.05}(y)$	0.96	0.95	0.94	0.86	0.61
VIB [1]	$\max_y p(y\|x)$ [34]	0.95	0.94	0.92	0.88	0.76
VIB [1]	$\log \mathrm{Dir}_{0.05}(y)$	0.97	0.96	0.94	0.88	0.78
NIBAE (Ours)	$\max_y p(y\|x)$ [34]	0.88	0.88	0.86	0.84	**0.81**
NIBAE (Ours)	$\log \mathrm{Dir}_{0.05}(y)$	0.90	0.95	0.92	0.86	0.80
NIBAE (Ours)	$+\log \mathcal{N}(z_n; 0, I)$	**0.98**	**0.99**	**0.99**	0.86	0.79

Table 5. Comparison of average corruption errors (%; lower is better) per severity level on CIFAR-10-C and CIFAR-100-C [32]. Bold and underline denote the best and runner-up results, respectively.

		CIFAR-10-C							CIFAR-100-C						
Architecture	Severity	Clean	1	2	3	4	5	Avg.	Clean	1	2	3	4	5	Avg.
ResNet-18	Cross-entropy	5.71	12.9	18.1	24.3	31.7	43.5	26.1	26.9	39.2	46.9	53.2	59.8	69.3	53.7
	VIB [1]	**5.47**	**12.5**	17.5	23.6	30.7	42.5	25.4	**26.5**	39.7	47.5	53.8	60.5	70.1	54.3
	NIBAE (Ours)	7.07	13.2	**17.2**	**21.7**	**27.5**	**37.0**	**23.3**	28.0	**39.0**	**45.5**	**51.4**	**57.6**	**67.0**	**52.1**
ViT-S/4	Cross-entropy	6.08	8.89	11.1	14.0	19.7	26.5	16.0	25.1	31.4	35.1	39.3	46.8	54.0	41.3
	VIB [1]	5.98	8.68	10.7	13.4	18.6	24.9	15.2	26.0	31.9	35.9	40.4	47.8	55.2	42.2
	AugMix [37]	6.52	8.97	10.8	13.4	18.4	23.9	15.1	24.9	29.9	33.3	37.1	43.6	51.1	39.0
	PixMix [38]	5.43	7.10	8.14	9.40	12.1	14.9	10.3	24.4	27.8	29.7	32.1	36.0	40.9	33.3
	NIBAE (Ours)	4.97	7.49	8.96	11.0	14.8	19.5	12.3	22.6	27.6	30.5	34.1	39.8	47.1	35.8
	+ AugMix [37]	5.35	7.65	8.99	11.0	14.2	18.4	12.0	**21.9**	26.4	29.1	32.4	37.8	44.3	34.0
	+ PixMix [38]	**4.67**	**5.90**	**6.55**	**7.45**	**9.12**	**11.4**	**8.08**	23.3	**26.0**	**27.5**	**29.3**	**32.6**	**36.5**	**30.4**

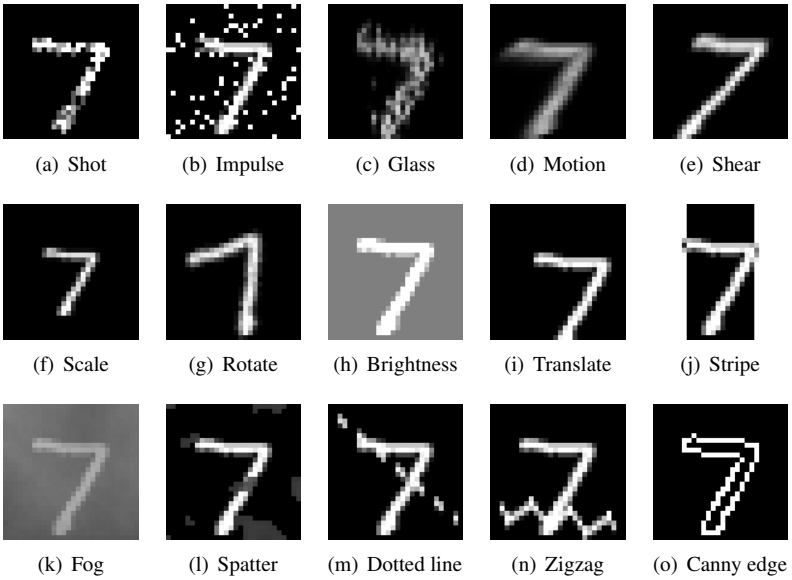
G.3. Robustness against adversarial examples

We evaluate adversarial robustness [27, 76, 93] adopting the *randomized smoothing* framework [14] that can measure a *certified* robustness for a given representation: specifically, any classifier can be "robustified" by averaging its predictions under Gaussian noise $\mathcal{N}(0, \sigma^2 I)$, where the robustness at input x depends on how consistent the classifier is on classifying $\mathcal{N}(x, \sigma^2 I)$ [47]. We adopt such a certified (or provable) protocol since it better aligns with our focus of testing robustness of representations that are not adversarially-trained [76]: empirical robustness, *i.e.*, that reports the worst-case accuracy after directly attacking a classifier with diverse adversarial attacks, is usually hard to get a non-trivial accuracy without a thorough adversarial training. The randomized smoothing based evaluation, on the other hand, provides a more meaningful metric for classifiers even for the "Cross-entropy" baseline, while still representing a lower-bound in robustness that a given classifier can achieve (with an aid of randomized smoothing) against *every* adversarial attack method.

We follow the standard certification protocol [14] to compare the *certified test accuracy at radius* r, which is defined by the fraction of the test samples that a smoothed classifier classifies correctly with its certified radius larger than r. We consider two scenarios for comparison: (a) *Standard*: the given models are directly smoothed out, and (b) *BatchNorm-adapted*: the models have information on the smoothing $\mathcal{N}(0, \sigma^2 I)$ *a priori*, and adapt their BatchNorm layers [42] to minimize the cross-entropy loss under $\mathcal{N}(x, \sigma^2 I)$ for 10 epochs of training samples. We use $\sigma = 0.1$ for this experiment.

Figure 3 summarizes the results: (a) for *Standard*, our proposed NIBAE achieves significantly better certified robust compared to the baselines at all radii tested. This confirms that the robustness of NIBAE is not only significant but also consistent *per input*, especially considering its high certified robustness at higher r's. Even in the case of *BatchNorm-adapted*, where the models already have prior knowledge on the threat model, NIBAE still maintains better feature extractors beyond BatchNorms and can improve the baselines in robust accuracies, *e.g.*, from $47.2 \rightarrow 51.4$ at $r = 0.25$.

G.4. Results on MNIST-C



(a) Shot (b) Impulse (c) Glass (d) Motion (e) Shear

(f) Scale (g) Rotate (h) Brightness (i) Translate (j) Stripe

(k) Fog (l) Spatter (m) Dotted line (n) Zigzag (o) Canny edge

Figure 5. Sample images in MNIST-C test dataset for different corruption types.

Table 6. Comparison of (a) clean error (%; lower is better), (b) AUROC on detecting Gaussian noise (higher is better), and (c) corruption errors (%; lower is better) per corruption type on MNIST-C [80]. Each classifier is trained on MNIST with random translation as augmentation. We highlight our results as blue whenever the value improves the baselines more than 3% in absolute values.

Method	*Clean*	*AUROC (↑)*	*Shot*	*Impulse*	*Glass*	*Motion*	*Shear*	*Scale*	*Rotate*	*Brightness*	*Translate*	*Stripe*	*Fog*	*Spatter*	*Dotted line*	*Zigzag*	*Canny edges*	*Average*
Cross-entropy	0.45	0.987	4.69	69.6	60.3	46.5	1.41	2.97	**4.80**	88.7	2.45	76.6	88.7	27.3	5.64	**27.3**	44.1	34.5
VIB [1]	**0.44**	0.988	4.52	73.5	73.8	71.8	1.73	2.84	5.85	90.1	2.15	78.1	89.8	28.4	5.85	28.5	44.0	37.6
sq-VIB [96]	0.48	0.955	4.32	71.5	63.5	62.3	1.62	2.70	5.74	90.5	2.43	80.3	90.3	24.8	5.91	32.0	43.4	36.4
NLIB [61]	1.15	0.974	7.13	67.9	62.5	57.9	2.15	4.00	7.06	**86.9**	3.28	81.8	88.7	30.1	8.97	31.0	41.8	36.4
sq-NLIB [96]	3.19	0.908	9.90	73.3	66.7	64.7	4.25	6.19	9.21	88.7	6.43	**72.4**	89.8	32.4	9.69	36.2	72.5	40.3
DisenIB [84]	0.54	0.997	4.60	68.8	56.4	50.4	1.11	**2.04**	4.84	88.7	2.01	74.3	**88.5**	20.1	4.75	27.4	69.0	35.2
NIBAE (Ours)	0.72	**1.000**	**3.71**	**48.8**	**44.0**	**27.1**	**0.99**	3.15	4.82	89.7	**0.88**	82.0	89.7	**16.4**	**4.14**	33.9	**25.9**	**29.8**

In this section, we evaluate our proposed NIBAE training on MNIST-C [80], a collection of corrupted versions of the MNIST [64] test dataset of 15 corruption types constructed in a similar manner to CIFAR-10/100-C [32], to get a clearer view on the effectiveness of our method on a simpler setup. For this experiments, we use a simple 4-layer convolutional network (with batch normalization [42]) as the encoder architecture, and trained every model on the (clean) MNIST training dataset for 100K updates following other training details of the CIFAR experiments (see Appendix B): again, we notice that the training does not assume specific prior on the corruptions. We compare NIBAE with the direct ablations of cross-entropy and VIB based models, as well as some variants of VIB, namely Nonlinear-VIB [61], Squared-VIB/NIB [96], and DisenIB [84]. Especially, we compare with DisenIB as (a) it considers a nuisance modeling (based on FactorVAE [54]) as NIBAE does, while (b) also tackling some robustness concerns, *e.g.*, its effectiveness on out-of-distribution detection for MNIST *vs.* Gaussian noise.

Table 6 summarizes the results: overall, we observe that the effectiveness of NIBAE training still applies to MNIST-C, *e.g.*, our NIBAE training improves the average corruption error from the baseline cross-entropy based training from

$33.1\% \rightarrow 29.8\%$, which could not be obtained by simply sweeping on the baseline VIB training. Given that MNIST-C allows a visually clearer distinction between contents and corruptions compared to CIFAR-10/100-C, one can better interpret the behavior of given models on each corruption types: here, we observe that our training can dramatically improve robustness for certain types of corruptions where the baselines shows poor performances, *e.g.*, Impulse, Glass, and Motion, while still some types of corruptions are still remaining challenging even with NIBAE, *e.g.*, especially for low-frequency biased corruptions such as Brightness and Stripe. Compared to DisenIB, on the other hand, we observe that the effectiveness from DisenIB, e.g., its gain in AUROC (as conducted in [84]), could not be further generalized to improve on MNIST-C, where NIBAE still improves upon it as well as achieving the perfect score at the same OOD task.

H. Application to model debugging

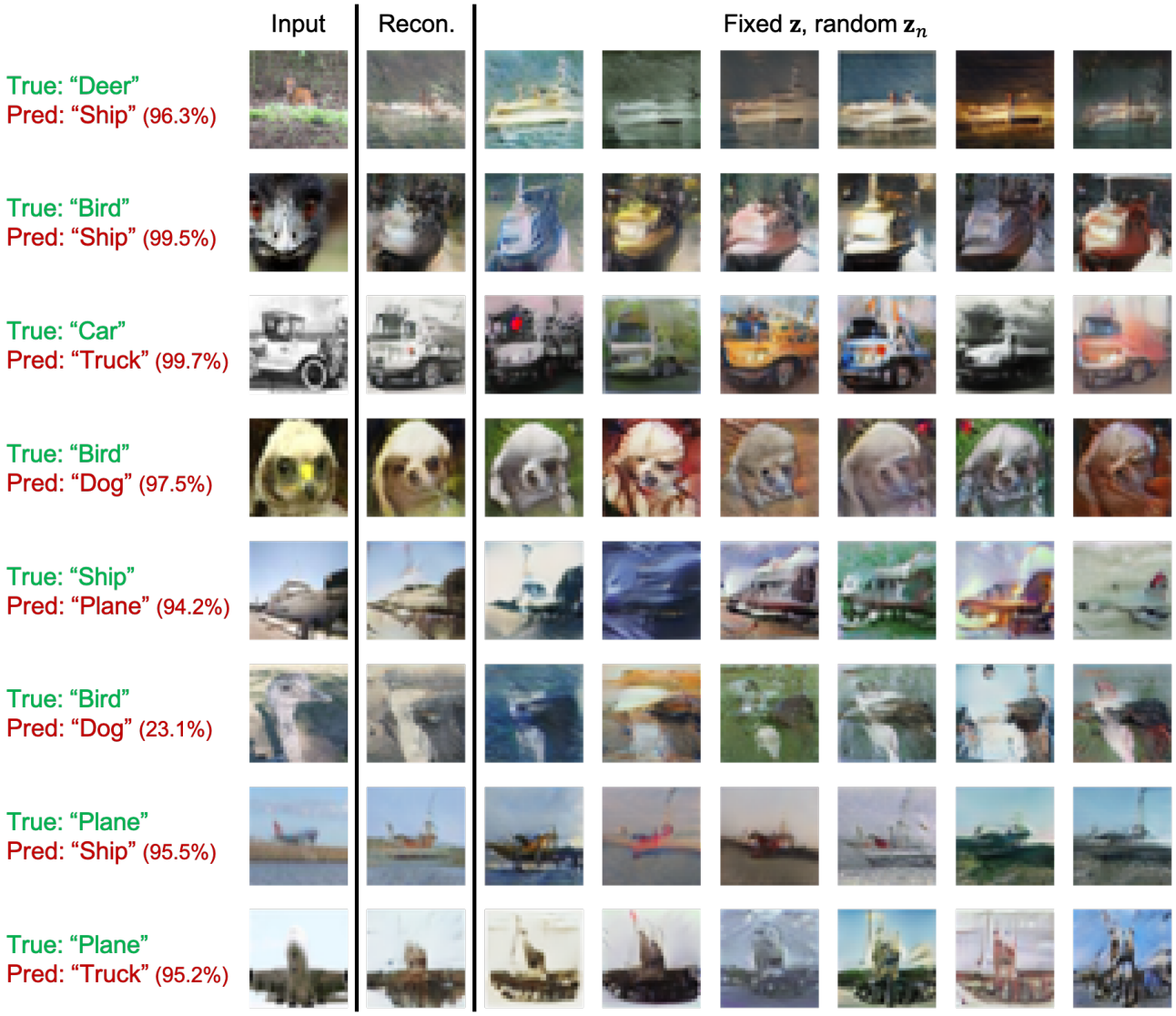


Figure 6. Qualitative comparisons between (a) the original input (the leftmost column), (b) its reconstruction (the second column), and (c) its further reconstructions with random nuisance $\mathbf{z}_n$ (the remaining columns), examined for test samples misclassified by a CIFAR-10 NIBAE model with ResNet-18 architecture.

To further understand how the proposed NIBAE model internally works with its representation $\mathbf{z}$ and $\mathbf{z}_n$, we examine an NIBAE model trained on CIFAR-10 to analyze how the model reconstruct given inputs when the model incorrectly classifies

them. Specifically, Figure 6 illustrates a subset of CIFAR-10 test samples misclassified by an NIBAE model by comparing the original input with its reconstructed samples from the model. Overall, we observe that such a qualitative comparison can provide a useful signal to interpret model errors: it effectively visualizes which visual cues of a given input negatively affected the decision making process of the given model, also visualizing the closest (misclassified) realizations that the model decodes for a given representation, *i.e.*, what the model actually perceived. For example, for the test input given at the first row of Figure 6, one can observe that the model essentially "ignored" the tiny part that represent the true semantic, *i.e.*, the "deer", and reconstructed the remaining part as a "ship".

I. Comparisons on image generation

I.1. Quantitative results

Table 7. Test FID (lower is better) and IS (higher is better) of GANs on CIFAR-10. Underline indicates the best. Value marked as * is reported from 2× longer training [50].

CIFAR-10, Uncond.	Augment.	FID (↓)	IS (↑)
StyleGAN2 [52]	HFlip	11.1	9.18
+ DiffAug [111]	Trans, CutOut	9.89	9.40
+ ContraD [48]	SimCLR	9.80	9.47
+ ADA [50]	Dynamic	**7.01***	-
+ FSD (R-18; Ours)	HFlip, Trans	8.43	9.68
+ FSD (R-50; Ours)	HFlip, Trans	<u>7.39</u>	**10.0**
FastGAN [69]	HFlip, Trans	34.5	6.52
+ Proj-GAN (R-18) [89]	HFlip, Trans	8.48	9.40
+ FSD (R-18; Ours)	HFlip, Trans	**7.80**	**9.65**

Table 8. Comparison of FID (lower is better) and IS (higher is better) of VAE-based models on unconditional generation. of CIFAR-10 and CelebA. Bold and underline denote the best and runner-up results, respectively.

Method	CIFAR-10		CelebA
	FID ↓	IS ↑	FID ↓
VAE [85]	115.8	3.8	-
VAE/GAN [85]	39.8	7.4	-
2s-VAE [16]	72.9	-	44.4
Perceptual AE [110]	51.5	-	13.8
NCP-VAE [3]	24.1	-	**5.25**
NVAE [101]	56.0	5.19	13.5
DC-VAE [85]	<u>17.9</u>	<u>8.2</u>	19.9
$L_{\texttt{recon}}$ (5) only	65.0	5.73	50.1
+ Adv. similarity (19)	46.8	6.29	25.1
+ Projection (R-18)	**12.6**	**8.86**	<u>6.91</u>

We also evaluate our proposed architecture and method as a *generative modeling*, especially focusing on the effectiveness of the *feature statistics encoder* (Section F) and the *adversarial similarity* based training (Appendix F) of autoencoders on CIFAR-10 [62] and CelebA [71] datasets. To this end, we consider an "unsupervised" version of NIBAE which omits the VIB loss ($L^{\beta}_{\texttt{VIB}}$; (14)) and the nuisance loss ($L_{\texttt{nuis}}$; (6)) in training, so that the model can assume an unconditional setup. Following other baselines, here we compute FIDs from 50,000 generated samples against the training dataset.

Table 8 summarizes the results. Firstly, it confirms the effectiveness of adversarial similarity based training: when it is solely applied upon $L_{\texttt{recon}}$ ("$L_{\texttt{recon}}$ only"; equivalent to [77]) it makes a significant improvements in both FID and IS. To further investigate the effectiveness of our proposed feature statistics encoder, we also test a scenario that the encoder is *fixed* by ResNet-18 pre-trained on ImageNet, akin to the setup of Table 7: we observe that our encoder design can surprisingly benefit from using better representation, *e.g.*, "+ Projection (R-18)" in Table 8 further improves FID on CIFAR-10 from $46.8 \rightarrow 12.6$, better than the best results among considered VAE-based models, by only training an MLP upon the feature statistics of the (fixed) model. It is notable that the gain only appears when we apply the adversarial similarity based training: *i.e.*, even with the pre-trained model, it only achieves 67.5 in FID on CIFAR-10 without the training. This observation suggests an interesting direction to scale-up autoencoder-based models by leveraging large, pre-trained representations, in a similar vein as [89] as presented in the context of GANs. Figure 8 illustrates uncurated samples generated from NIBAE with Projection trained on CelebA.

I.2. Qualitative results

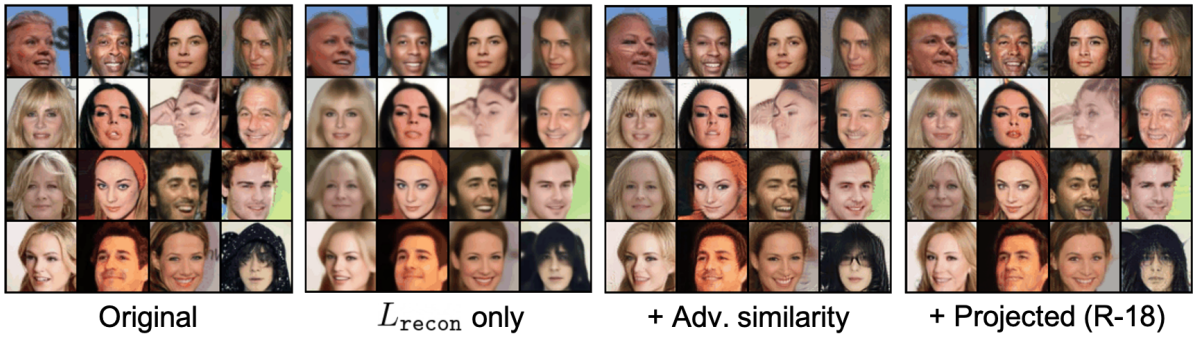
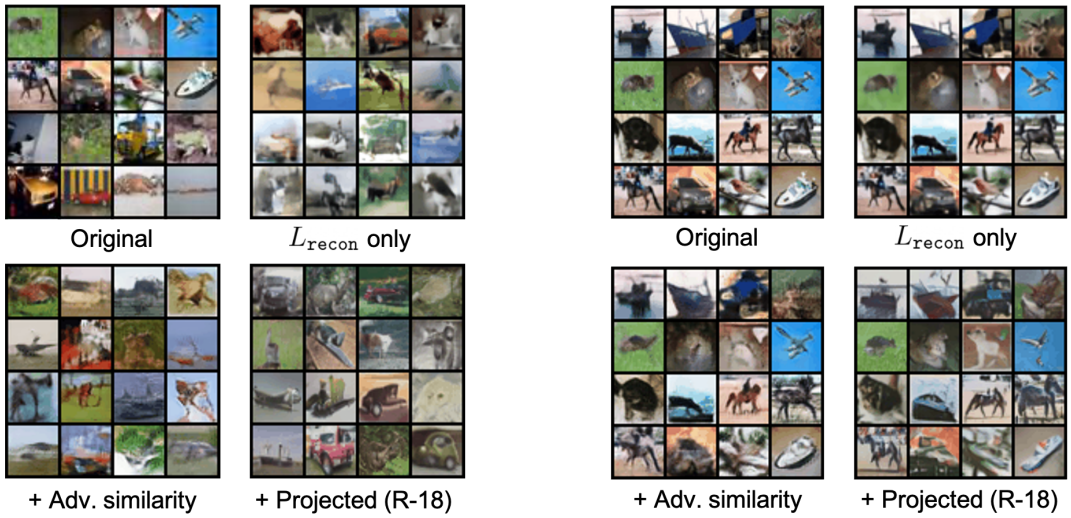


Figure 7. Qualitative comparison on reconstructions from fixed samples of unconditional NIBAE model (and its ablations) on CelebA.

(a) $L_{\texttt{recon}}$ only (FID: 50.1) (b) + Adv. similarity (FID: 25.1) (c) + Projected (FID: 6.91)

Figure 8. Qualitative comparison on uncurated random samples generated from unconditional NIBAE model (and its ablations) on CelebA.



(a) Reconstruction (b) Sampling

Figure 9. Qualitative comparison on uncurated random samples from unconditional NIBAE model (and its ablations) on CIFAR-10.