Synthesize-It-Classifier: Learning a Generative Classifier through Recurrent Self-analysis

Arghya Pal, Raphaël C.-W. Phan, KokSheik Wong
School of Information Technology, Monash University, Malaysia campus
arghya.pal@monash.edu

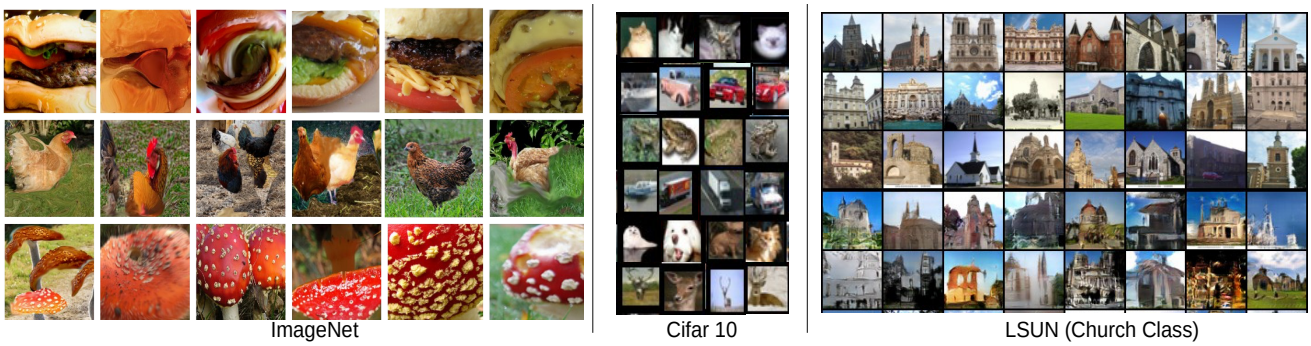

Figure 1: ***Qualitative Results of STIC Method:*** *(best viewed while zoomed-in):* We show qualitative results on the ImageNet [5], CIFAR 10 [18] and LSUN [34] datasets. (1) ImageNet: we show results of cheeseburger, chicken and mushroom classes. (2) CIFAR 10: the STIC synthesises photo-realistic images of cat, automobile, frog, truck, dog and deer classes (top-bottom rows). Variation of style (illumination, background) and content (pose, shape) can be seen for each of the classes. (3) LSUN: We show visible geometric regularities in house shape, dome-like structures, and other outdoor entities (sky, illumination). All images are generated by using $\tau = 10$ passes. The STIC methodology description is in Sec. 3.

Abstract

We show the generative capability of an image classifier network by synthesizing high-resolution, photo-realistic, and diverse images at scale. The overall methodology, called Synthesize-It-Classifier (STIC), does not require an explicit generator network to estimate the density of the data distribution and sample images from that, but instead uses the classifier's knowledge of the boundary to perform gradient ascent w.r.t. class logits and then synthesizes images using the Gram Matrix Metropolis Adjusted Langevin Algorithm (GRMALA) by drawing on a blank canvas. During training, the classifier iteratively uses these synthesized images as fake samples and re-estimates the class boundary in a recurrent fashion to improve both the classification accuracy and quality of synthetic images. The STIC shows that mixing of the hard fake samples (i.e. those synthesized by the one-hot class conditioning), and the soft fake samples (which are synthesized as a convex combination of classes, i.e. a mixup of classes [36]) improves class interpolation. We demonstrate an Attentive-STIC network that shows iterative drawing of synthesized images on the ImageNet dataset that has thousands of classes. In addition, we introduce the synthesis using a class conditional score classifier (Score-STIC) instead of a normal image classifier and show improved results on several real world datasets, i.e. ImageNet, LSUN and CIFAR 10.

1. Introduction

Discriminative classifiers $p(y|x)$ and generative models $p(x)$ are conventionally considered as domains complementary to each other, yet the distinction between them is blurring. A generative model $p(x)$ [26] appears as a data generation process that captures the underlying density of a data distribution, whereas the discriminative classifier learns complex feature representations of images with a view to learn the class boundaries for subsequent classification. There is a recent growth of interest in Machine Learning (ML) and Computer Vision (CV) [15, 20, 7] to use a discriminative classifier and then synthesize novel samples from its understanding of class boundary information. To elaborate, in the model of [7], the classifier $p(y|x)$ log-

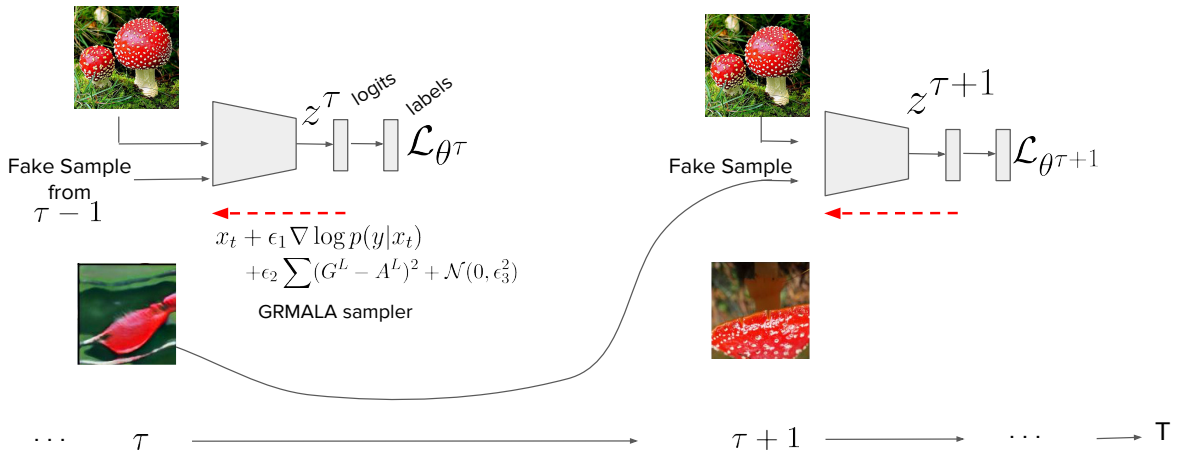

Figure 2: ***The STIC Methodology:*** Our main objective is to learn a class conditional model by emphasizing the fact that $p(x|y) \propto p(y|x)$, Eq 1, and synthesise photo-realistic images from a discriminative classifier. Our proposed STIC serves dual objectives: (1) learning smooth class boundaries with Vicinal Risk Minimization; and (2) learning tighter class boundaries using recurrent self-analysis class boundary re-estimation. At time $(\tau + 1)$, the classifier $p(y|x)$ is adjusting the parameters $(\theta^{\tau+1})$ using real images, mixup images; and in addition to that, synthesized images from real classes and synthesized images from mixup classes from previous iteration τ (marked as Fake Sample) are provided to the classifier. Please note that mixup classes are not actually classes but the mixup of logits of two or more classes. The samples are, at time (τ), synthesized from classifier's knowledge of the class boundary by gradient ascending w.r.t class logits, z^{τ}, using our proposed Gram Matrix Regularized Metropolis Adjusted Langevin Algorithm sampler (GRMALA), see red dashed arrow. The STIC discriminative classifier is trained for $\tau \in \{1, 2, \cdots, \mathrm{T}\}$ number of iterations.

its are used to estimate the joint density of the image-label $p(x, y)$, and the marginal of the image distribution, $p(x)$; note the random variables, x: image, and y: class label. Meanwhile in [15, 20] the classifier logits are used to produce synthesized samples using an MCMC-like sampling mechanism. The classifier, on the other hand, tries to distinguish these synthesized samples and the real images to re-estimate class boundaries. We remark that synthesizing novel samples from a discriminative classifier hinges on an important factor: how well the discriminative classifier has learned the class boundaries?

We note that all the discriminative classifiers in [7, 15, 20, 22] used for synthesizing novel samples are trained with Empirical Risk Minimization (ERM) [31]. Yet, from the literature [4, 36], it is evident that a discriminative classifier trained with ERM does not provide a smoother estimate of uncertainty near to the class boundary regions [4]. Hence, we ask ourselves the question: *does training with ERM have any consequence on the synthesizing capabilities of these discriminative classifiers?* We note that the transitions at class interpolation and sample quality towards class boundaries of these discriminative classifiers are neither smooth nor photo-realistic. In this work, we primarily seek to address this problem, *viz.*, to build a discriminative classifier that will serve dual objectives: (1) the interpolated samples from one class to another must be photo-realistic; and (2) the classifier must learn tighter class boundaries so as to generate photo-realistic samples.

To address the first objective, we train the discriminative classifier with Vicinal Risk Minimization (VRM) [36]. We leverage more virtual mixup image-label samples [36] in addition to the real image-label samples and train the classifier. We then synthesize novel samples. Our novel sample synthesis method is, by design, similar to the Style Transfer work [6], i.e. starting with an initial image x_0 which is updated with gradient ascent using our proposed novel Gram Matrix Regularized Metropolis Adjusted Langavin Algorithm (GRMALA) sampler. To the best of our knowledge, this is the first discriminative classifier trained with VRM and subsequently synthesized using a novel GRMALA sampler. We will discuss this in detail in Sec 3.

Training a discriminative classifier with VRM alone, is however, a necessary condition for learning the smoother estimation of uncertainty among classes, but not a sufficient condition that provides tighter class boundaries. Cognitive studies [1, 3] have shown evidence where subjects (i.e. human) start with a weak cognitive decision model of an environment or the world, and recurrently refine through mistakes and self-analysis gained from the environment to develop much stronger cognitive decision models. In a similar spirit, we present our recurrent discriminative network trained with VRM, that we call Synthesize-it-

Classifier (STIC). The STIC recurrently eliminates the regions which are outside of the class boundaries and forces the sampler to search within class boundaries. The STIC methodology trains the classifier with real images of different classes and then synthesizes samples conditioned on a class as well as the mixup samples w.r.t. the class logits. At the next pass, the STIC inputs these synthesized samples as fake samples to the already trained discriminative classifier of the previous pass, thus allowing the classifier to re-estimate class boundaries using real images, the synthesized mixup images and the synthesized samples (we call this self-analysis). Similar to [15, 20], we are, in a way, asking the classifier to quantify its own generated samples with respect to the class boundaries. The STIC does the recurrent self-analysis for $\tau \in \{1, 2, \cdots, T\}$ number of passes.

From our empirical observations, we note that if the image space is large (typically $> 227 \times 227$), the GRMALA sampler exhibits a slow update. We hence show an attentive-STIC where the discriminative classifier operates on the feature space instead of raw pixel space, thus exhibiting fast update. Additionally, we also propose a novel class conditional score matching based discriminative classifier that matches the derivative of the model's density with the derivative of the data density [29]. We will discuss each of these components elaborately in Sec 3.

Our contributions can be summarized as follows:

- Novel recurrent self analytic STIC trained with VRM and show synthesized images using Gram matrix Regularized MALA (GRMALA) sampler w.r.t class logit
- We show Attentive-STIC model to address the slow mixing problem of MALA-approx. We also propose a novel class conditional score function based discriminative classifier (we call it the Score-STIC method)
- We show results on several real world datasets, such as ImageNet, LSUN and CIFAR 10

2. Related Work

Generative Discriminative Learning: The generative classifier methodology was first evident in the seminal paper "self-supervised boosting" [32], that learns a sequence of weak classifiers using the real data and self-generated negative samples. The use of negative samples while learning in an unsupervised manner is also seen in [12]. Similar to that, the methods in [15, 20] use the Convolutional Neural Network (CNN) based discriminative classifier's logits and produces synthesized samples using an MCMC-like sampling mechanism. The classifier tries to distinguish these synthesized samples and the real images to learn class boundaries. Similar to those lines of work, the method [21] shows that learning class boundaries from real and synthesized images is equivalent to optimizing the Wasserstein distance between real image and synthesized image density. Recently, the work [7] shows learning of a joint distribution and a marginal distribution from the knowledge of the class logits of a discriminative classifier.

Style Transfer: There is a plethora of works that perform style transfer to meet various alternate objectives, such as: a generative adversarial learning approach to disentangle style and content of an image [16]; while [17] propose to capture the particularity in style, and the capturing style and content of an image. The style disentanglement is shown for single image super resolution in [37]. However, in this work we will use the Gram Matrix based style transfer proposed in [6]. The seminal work of [6] computes a Gram Matrix, $G^L \in \mathbb{R}^{N_L \times N_L}$ using the following: the L^{th} layer of a Convolutional Neural Network (CNN) has distinct N_L feature maps each of size $M_L \times M_L$. The matrix $F^L \in \mathbb{R}^{N_L \times M_L}$, stores the activations $F^L_{i,j}$ of the i^{th} filter at position j of layer L. Then, the method computes feature correlation using: $G^L_{i,j} = \sum_k F^L_{i,k} F^L_{j,k}$, where any $F^m_{n,o}$ conveys the activation of the n^{th} filter at position o in layer m.

Metropolis-adjusted Langevin algorithm (MALA): The Metropolis-Hastings (MH) [23] uses the transition operator, *viz.* $x_{t+1} = x_t + \mathcal{N}(0, \epsilon_1^2)$, $\alpha = p(x_{t+1})/p(x_t)$, and if $\alpha < 1$ reject the sample x_{t+1} with probability $(1 - \alpha)$ and set $x_{t+1} = x_t$ else keep x_{t+1}. In practice, the MH is very slow to produce samples from any computable distribution $p_{data}(x)$. As a remedy, [27, 28] have proposed an approximation method called Metropolis-adjusted Langevin algorithm, or the MALA. Starting with an initial x_0 typically sampled from a Gaussian distribution $\mathcal{N}(0, I)$, the MALA uses the transition operator, *viz.* $x_{t+1} = x_t + \frac{1}{2\sigma}\nabla \log p(x_t) + \mathcal{N}(0, \sigma^2)$, $\alpha = f(x_{t+1}, x_t, p(x_{t+1}), p(x_t))$, and if $\alpha < 1$ reject x_{t+1} else keep x_{t+1}, and samples from the distribution $p(x)$. The method [25] uses the stochastic gradient Langevin dynamics (SGDL) to get rid of the rejection steps of MALA and proposed the MALA-approx method. In addition to that, the method [25] uses different step sizes ϵ_1 and ϵ_2 in: $x_{t+1} = x_t + \epsilon_1 \nabla \log p(x_t) + \mathcal{N}(0, \epsilon_2^2)$ and exhibits more control over variability. In this work, we will propose a novel Gram Matrix Regularized MALA and the sampler takes the form: $x_{t+1} = x_t + \epsilon_1 \nabla \log p(x_t) + \sum (G^L(x_t) - A^L(x_t))^2 + \mathcal{N}(0, \epsilon_2^2)$, where ϵ_1 and ϵ_2 are scaling factors.

Vicinal Risk Minimization (VRM) using Mixup: The Empirical Risk Minimization (ERM) [31] learns a function $f \in \mathcal{F}$ that determines the non-linear relation of the image samples $x_i|_{i=1}^N$ and the corresponding classes $y_i|_{i=1}^N$ sampled from a data distribution $p_{data}(x, y)$ by optimizing the empirical risk, $R(f) = \frac{1}{N}\sum_{i=1}^N l(f(x_i), y_i)$. The loss function $l(\cdot)$ can be any standard loss function. Learning the function f by minimizing ERM leads the function f to memorize the training samples instead of a good gen-

eralization even under the purview of strong regularizer [4]. To mitigate this, [4] proposed an alternate risk minimization technique known as Vicinal Risk Minimization (VRM), i.e. $R_{vicinity}(f) = \frac{1}{N+M}\sum_{k=1}^{N+M} l(f(\hat{x}_k), \hat{y}_k)$. In VRM, we augment additional image-label pairs $(\tilde{x}_i, \tilde{y}_i)|_{i=1}^{M}$ using simple geometric transformations (such as crop, rotation, mirror) of real image-label pairs $(x_i, y_i)|_{i=1}^{N}$. We get the set of image-labels $(\hat{x}_k, \hat{y}_k)|_{k=1}^{N+M}$ comprising augmented image-label and the real image-label pairs. The Mixup [36] extends this idea by augmenting virtual image-target samples, $x_k^{mixup} = \lambda x_i + (1-\lambda)x_j$ and $y_k^{mixup} = \lambda y_i + (1-\lambda)y_j$, where $\lambda \sim \text{Beta}(\alpha, \alpha)$ is sampled from Beta distribution, for $\alpha \in (0, \infty)$, also x_i, x_j, and y_i, y_j are real image-labels. Mixup shows results by combining real image-label samples of different classes instead of hand-crafted data augmentation of images. The VRM of Mixup can be defined as, $R_{mixup}(f) = \frac{1}{N+K}\sum_{l=1}^{N+K} l(f(x_l), y_l)$. We get the set of image-labels $(x_l, y_l)|_{l=1}^{N+K}$ from the real image-label pairs and mixup image-label pairs, and will use these in this work.

3. The STIC Methodology

In this work, we wish to learn the parameter of a class conditional distribution of image x and the corresponding class label y (we fix y to be from a particular class y_c), i.e.:

$$p(x|y = y_c) \tag{1}$$

with a view to generating photo-realistic novel samples.

We expand the class conditional model in Eq 1 using the Bayes rule, i.e.: $p(x|y) = p(x)p(y|x)/p(y) \propto p(x)p(y|x)$. We however cannot directly write a sampler by utilizing the "product of experts" [13], as we do not have a generator network $p(x)$ in our setup. Since the random variable y is categorical, we instead can write a modified version, i.e:

$$p(x|y) = p(x)p(y|x)/p(y) \propto p(y|x) \tag{2}$$

such that, estimating the density directly has a relation to how well synthesized samples are classified by the discriminative classifier network.

Following the Style Transfer work [6] and the sampling with Langevin algorithm work in [25, 27], we propose a Gram Matrix Regularized MALA approx (GRMALA) sampler and propose the following update rule for x_{t+1}:

$$x_t + \epsilon_1 \nabla \log p(y|x_t) + \epsilon_2 \sum (G^L - A^L)^2 + \mathcal{N}(0, \epsilon_3^2) \tag{3}$$

and, similar to MALA-approx proposed in [25], we use different step sizes, i.e. ϵ_1, ϵ_2, and ϵ_3 for three terms after x_t in Eq 3. Here, ϵ_1 and ϵ_2 control the sample quality and ϵ_3 controls the diversity by moving around the search space. Note that we get the Gram Matrix G^L from the x_t and we get Gram Marix A^L from a real image x (for more details on Gram Matrix please refer to [6], or Sec 2 Style Transfer section). In order to generate photo-realistic synthsized images, our discriminative classifier, hence, must serve two objectives: (1) *Learning Smooth Class Boundaries using VRM* such that the interpolated samples from one class to another must be photo-realistic; and (2) *Learning of Tighter Class Boundaries using Recurrent Self-analysis Class Boundary Re-estimation* such that the classifier must learn tighter class boundaries so as to generate photo-realistic samples.

Learning Smooth Class Boundaries using VRM: Similar to [36], we augment mixup image-label pairs along with real image-label pairs. We have K number of mixup augmented image-label pairs $(x_k^{mixup}, y_k^{mixup})|_{k=1}^{K}$, those we get after, $x_k^{mixup} = \lambda x_i + (1-\lambda)x_j$ and $y_k^{mixup} = \lambda y_i + (1-\lambda)y_j$, where $\lambda \sim \text{Beta}(\alpha, \alpha)$, for $\alpha \in (0, \infty)$, also x_i, x_j, and y_i, y_j are real image-label pairs. For brevity, let us assume that the mixup image-label pairs are coming from a mixup distribution $(x_k^{mixup}, y_k^{mixup}) \sim p_{mixup}(x^{mixup}, y^{mixup})$ and we have our real image-label distribution $(x_i, y_i) \sim p_{data}(x, y)$. Our objective function to optimize Eq 2 is the following:

$$\begin{aligned}\mathcal{L}(\theta) = &- \sum_{(x_i, y_i) \sim p_{data}}^{i=1,\cdots,N} \log p_\theta(y_i = y_c | x_i) \\ &- \sum_{(x_k^{mixup}, y_k^{mixup}) \sim p_{mixup}}^{k=1,\cdots,K} \log p_\theta(y_k = y_{mixup} | x_k^{mixup})\end{aligned} \tag{4}$$

where we note here that $y_k = y^{mixup}$ is not a true class but represents the mixing of true class logits.

Learning of Tighter Class Boundaries using Recurrent Self-analysis Class Boundary Re-estimation: Learning smooth class boundaries using VRM is a necessary condition for smooth image synthesis but not a sufficient condition for learning tighter class boundaries with a view to synthesize photo-realistic images. We hence introduce a recurrent self-analysis class boundary re-estimation methodology that eliminates the regions which are outside of the class boundaries and force the sampler to focus within the class boundaries. To achieve this objective, we now describe a recurrent training procedure that spans around $\tau \in \{1, 2, \cdots, \text{T}\}$ number of passes. At pass τ, we synthesize novel samples from a trained classifier $p_\tau(\cdot)$ by GRMALA based update with respect to the class logits. At the next pass, $\tau + 1$, the STIC takes images from dataset and mixup images as real images. On the other hand, synthesized images of real classes and synthesized images of mixup classes from the classifier at pass τ are taken as fake samples (note that such synthesized samples are taken from the trained

classifier at previous pass τ, see Fig 2 fake images). Thus allowing the classifier to re-estimate class boundaries using the real images, the synthesized mixup images and the synthesized samples. We call this a recurrent self-analysis. The recurrent class boundary re-estimation is, in a way, asking the classifier to quantify its own generated samples with respect to the class boundaries. We sample and re-train the classifier for $\tau \in \{1, 2, \cdots, \mathrm{T}\}$ times, thus enabling the classifier to re-estimate its class boundaries at each time step. For the $(\tau+1)^{th}$ time step, the objective function of the classifier hence then becomes:

$$\begin{aligned}\mathcal{L}(\theta^{\tau+1}) = &- \sum_{(x_i,y_i)\sim p_{data}}^{i=1,\cdots,N} \log p_{\theta^{\tau+1}}(y_i = y_c|x_i) \\ &- \sum_{(x_k^{mixup},y_k^{mixup})\sim p_{mixup}}^{k=1,\cdots,K} \log p_{\theta^{\tau+1}}(y_k = y_{mixup}|x_k^{mixup}) \\ &- \sum_{(x_i,y_i)\sim p_{\theta^\tau}}^{i=1,\cdots,N} \log p_{\theta^{\tau+1}}(y_i = -1|x_i) \\ &- \sum_{(x_k^{mixup},y_k^{mixup})\sim p_{\theta^\tau}}^{k=1,\cdots,K} \log p_{\theta^{\tau+1}}(y_k = -1|x_k^{mixup})\end{aligned} \tag{5}$$

Theoretically, the softmax of the classifier $p_\theta^{\tau+1}(y|x)$ is $\frac{\exp(p_\theta^{\tau+1}(x)[y])}{\sum_{y'}\exp(p_\theta^{\tau+1}(x)[y'])}$. Thus, we can approximate $p(x, y)$ as $p_\theta^{\tau+1}(x, y) = \exp(p_\theta^\tau(x)[y])/Z(\theta)$, where $p_\theta^\tau(\cdot)$ is from the previous time step τ. Marginalizing y from $p_\theta^{\tau+1}(x, y)$, i.e. $p_\theta^{\tau+1}(x) = \sum_y p_\theta^{\tau+1}(x, y) = \sum_y \exp(p_\theta^\tau(x)[y])/Z(\theta)$ provides the estimation of $p(x)$. However, $p(x)$ is dropped from Eq 2 as there is no explicit network and learning is incorporated through GRMALA and $p_\theta^\tau(\cdot)$.

4. Experiments and Results

We perform a detailed suite of experiments and ablation studies, across standard benchmark datasets; notably : ImageNet [5], Cifar 10 [18] and LSUN [34].

Baseline and SOTA methods: By design, our method is a hybrid network that can simultaneously perform classification and synthesis. From the class conditional generative network end, we observe that the BigGAN [2], PnP [25], SNGAN [24] methods are state-of-the-art (SOTA) for class conditional image generation. In terms of the generative discriminative learning, the works of JEM [7], INN [15], WINN [22], EBM [33] are closer to our work. However, our proposed STIC, to a large extent, differs from these methods as follows: (1) the crucial difference is that our discriminative classifier is trained with VRM, and (2) we use a novel Gram Matrix MALA sampler. We consider BigGAN-deep (res 256, channel 96, parms 158.3, shared, orthogonal reg, skip-z) [2], cascade classifier network model from [15, 22] methods, and other methods as described in their paper. For classifiers, we consider ResNet [10], MobileNet [14], and GoogleLenet (GLent) [30] as the SOTA methods to compare our method against. We consider INN [20] as our baseline for synthesizing method, as we note that such earlier effort uses a discriminative classifier to synthesize novel samples from its understanding of class boundary information. These synthesized samples and real images are then utilized by INN method for class boundary re-estimation. For discriminative classifier, we use GoogleLeNet as our baseline method. Here, a batch size of 50 is considered for all SOTA methods unless specified otherwise.

Network Setup and Hyperparameter Choices of STIC: Similar to the previous work [7], we use a Wide Residual Network [35], WideResNet-28-10, without batch-normalization to make STIC output deterministic functions of the input. The Adam optimizer, $5k$ iteration for each pass $\tau \in \{1, 2, \cdots, 10\}$ totaling $50k$ iterations, the Langevin dynamics chains are evolved after 15 epochs (after one pass) and with probability 0.5 we re-initialize the chains with uniform random noise. We have two notions for time, a pass τ and iteration: we start training, at pass $\tau = 1$. At pass $\tau = 1$, the classifier trained with real images and virtual mixup images. At that time, we consider blank images (pixel intensities are set to 255) as fake images (i.e. $y_i = -1$ in Eqn 4). One pass continues for $5k$ iterations and then we synthesize fake images from the classifier $p_{\theta^1}(y|x)$. We then move to the next pass $\tau = 2$ that lasts for another $5k$ iterations. We have a total number of 10 passes, i.e. $50k$ iterations, for STIC training.

Qualitative Results: Sample labeled image generations of the proposed STIC method are summarized in Fig 1 and more images are in Suppl. mat. Note that STIC generates images with improved quality in multiple cases across the datasets. In LSUN, proper geometric shapes are observed for house and sky of synthesized images by STIC. In ImageNet and in Cifar 10 synthesized images, we observe that style and content information are captured by STIC.

Diversity Analysis: At pass τ, we synthesize class conditioned sample $p_{\theta^\tau}(x|y = y_{c_1})$ of class y_{c_1} (see black arrows in Fig 3(a)). Similar to marginal density estimation proposed in [7], we use a small neighborhood around $p_{\theta^\tau}(x|y = y_{c_1})$ as other starting samples to understand the capability of the model to generate diverse samples. It is evident that samples which are in near vicinity show similar object appearance (observe same face structures of black arrow samples in Fig 3(a)), similar background (observe similar facial structure and background in red arrow samples in 3(a)). In contrast, samples which are far apart, for example, see red arrow and purple arrow samples in 3(a), show different appearance of the same dog class.

Latent Space Interpolation: Two points $p(x|y = y_{c_1})$ and

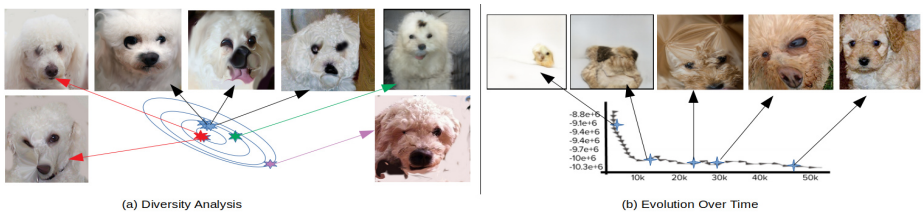

Figure 3: ***(a) Diversity Analysis:*** we synthesize samples from one class and samples from the neighborhood around those samples to get other starting samples on ImageNet class dog. We note, samples which are in near vicinity show similar object appearance (observe same face structures of black arrow samples). In contrast, samples which are far apart (see red arrow and purple arrow samples) show different appearance of the same dog class. ***(b) Evolution Over Time:*** we show class dog synthesized samples of ImageNet at different iterations, i.e. $\{10k, 20k, \cdots, 50k\}$ (horizontal axis: no. of iterations, vertical axis: training loss). Images are blurry initially but become clearer over time, showing that the proposed method is learning tighter class boundaries over the time steps.

Figure 4: *(a)* ***Image Interpolation*** *(ours)* first four columns show image interpolation result of our method. We notice smooth transition from one class c_1 to another class c_2. *(b)* ***Interpolation of Result of INN*** *(baseline):* We note that the class interpolation from one class to other is not smooth, i.e. in-between images are not human interpretable.

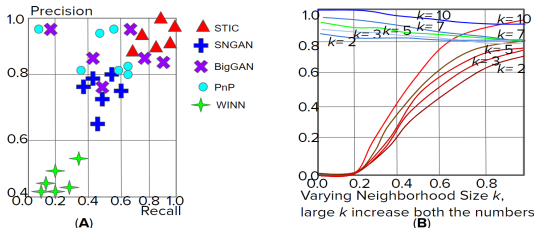

Figure 5: **Generalizability of STIC Method: (A)** We show the precision-recall comparison of STIC, SNGAN, BigGAN, PnP and WINN at different initializations. A high precision-recall for STIC justifying our claim. **(B)** Precision-recall at different k-NN using the features of a ResNet50 classifier.

$p(x|y = y_{c_2})$ are sampled from two distinct classes c_1 and c_2 at pass $\tau = 10$, and linearly interpolated between $p(x|y = y_{c_1})$ and $p(x|y = y_{c_2})$ to obtain novel samples. The synthesized images of ImageNet are shown in Fig 4(a). Synthesized images from one class to another are smooth and human interpretable, as opposed to the interpolation provided by the baseline INN [15] in Fig 4(b), i.e. in-between images are not human interpretable. This supports our claim that STIC provides smooth synthesised samples.

Evolution over Time Steps: In Fig 3(b), we show the qualitative results of class dog of ImageNet at different iterations, i.e. $\{10k, 20k, \cdots, 50k\}$. Please note that, in STIC setup, $5k$ iteration stands for one pass of $\tau \in \{1, 2, \cdots, 10\}$. The generated images are blurry initially but become clearer over time, showing that the proposed method is learning tighter class boundaries over the time steps.

Quantitative Evaluation: We used multiple quantitative metrics to study the proposed method on generated image quality, diversity and image-label correspondence: (i) **MIS** (↑, higher is better) [9]; (ii) **FID** (↓, lower is better) [11]; (iii) $\textbf{Cls}_R$(↑, higher is better), i.e. Top-5 classification accuracy (in %) of a ResNet-50 classifier trained on real labeled images and tested on generated images; and (iv) $\textbf{Cls}_G$(↑, higher is better), i.e. Top-5 classification accuracy (in %) of a ResNet-50 classifier trained on generated/synthesized labeled images and tested on real images. The results are shown in Table 1. We observe a distinct performance gain for STIC over the state-of-the-art models. The low FID score and high $\textbf{Cls}_G$-based classification accuracy scores imply diverse image-label generation. In particular, the improved classification performance, as shown through $\textbf{Cls}_T$ and $\textbf{Cls}_G$, demonstrate the utility of the synthesized labeled images for downstream classification tasks.

Classification Accuracy improvement with STIC: In Table 2, we show that STIC improves not only the generation quality (as shown in Fig 1 but also the classification accuracy. We tried to show through $\textbf{Cls}_G$ that, training four classifiers ResNet, WideResNet, MobileNet, GoogleLeNet (shown

Methods	LSUN				CIFAR10				ImageNet			
	MIS (↑)	**FID** (↓)	**Cls**$_R$ (↑)	**Cls**$_G$ (↑)	**MIS** (↑)	**FID** (↓)	**Cls**$_R$ (↑)	**Cls**$_G$ (↑)	**MIS** (↑)	**FID** (↓)	**Cls**$_R$ (↑)	**Cls**$_G$ (↑)
INN	14.91	45.62	26	10	0.93	118.92	29	20	1.92	189.05	52	30
WINN	17.43	38.03	41	28	21.94	51.81	48	36	21.13	58.72	48	38
PnP	32.03	15.07	62	58	31.37	17.93	54	53	33.18	14.71	61	54
JEM	28.92	40.42	60	39	38.4	47.60	57	39	32.32	40.41	53	32
EBM	31.83	19.73	62	50	31.63	17.02	58	50	32.81	30.90	63	52
BigGAN	**113.13**	**8.67**	88	87	100.31	**7.92**	89	81	99.31	**8.51**	85	80
SNGAN	52.37	17.43	61	59	53.01	20.3	83	78	65.72	12.62	67	61
STIC	93.61	13.32	96	92	97.91	12.81	91	90	98.62	15.01	95	93
STIC-ERM	30	35.92	72	62	20	48.17	61	60	27.19	38.27	65	63
Attentive-STIC	99.61	9.01	97	95	100.56	11.71	93	90	100.19	10.38	96	93
Score-STIC	112.61	8.82	**98**	**96**	**108.62**	9.99	**97**	**92**	**104.91**	8.83	**97**	**95**

Table 1: **Quantitative Results of Various Real-world Image Datasets:** We report: (i) **MIS** (↑, higher is better); (ii) **FID** (↓, lower is better); (iii) **Cls**$_R$(↑, higher is better); and (iv) **Cls**$_G$(↑, higher is better). We mark winning entries in bold. The STIC and its variants are underlined. The N/A stands for not applicable.

Cls$_G$(↑)	INN	WINN	PnP	JEM	EBM	BigGAN	SNGAN	**STIC**
LSUN	20/29 /10/18	23/31 /38/19	62/70 /80/73	60/70 /80/72	22/22 /19/17	41/40 /38/35	55/50 /51/38	58/50 /50/41
CIFAR10	10/08 /14/07	19/10 /05/09	50/46 /49/50	55/50 /49/49	16/19 /10/09	52/50 /53/51	58/60 /59/59	60/62 /58/59
ImageNet	05/02 /03/04	07/03 /02/03	38/30 /36/30	41/30 /37/30	20/18 /10/19	53/50 /69/71	51/56 /57/55	60/76 /75/70
Cls$_R$(↑)	INN	WINN	PnP	JEM	EBM	BigGAN	SNGAN	**STIC**
LSUN	10/13 /11/12	18/18 /17/19	61/62 /68/63	53/59 /58/52	22/22 /19/17	41/40 /33/30	55/50 /50/38	62/59 /56/72
CIFAR10	06/06 /04/06	09/07 /04/09	45/43 /43/40	45/40 /43/43	10/11 /10/07	42/40 /43/49	46/56 /58/57	57/63 /74/73
ImageNet	05/02 /03/04	07/03 /02/03	38/30 /36/30	41/30 /37/30	20/18 /10/19	43/40 /39/31	43/39 /31/30	63/63 /83/80

Table 2: **Classification Accuracy Improvement with STIC:** We report: (i) **Cls**$_R$(↑, higher is better); and (ii) **Cls**$_G$(↑, higher is better). Each cell of the table shows classifier accuracy of ResNet/ WideResNet/MobileNet/GoogleLeNet/STIC

ResNet/WideResNet/MobileNet/GoogleLeNet in Fig 2) *purely* using generated images of INN, WINN, PnP, JEM, EBM, BigGAN and SNGAN reduces the classification accuracies on CIFAR10, ImageNet and LSUN dataset. But, STIC has shown an improved result. For **Cls**$_R$, we trained ResNet, WideResNet, MobileNet, GoogleLeNet on real images and tested on INN, WINN, PnP, JEM, EBM, BigGAN and SNGAN generated images. This shows that the recurrent self-analysis obtains tighter class boundaries. For example: ResNet/ WideResNet/MobileNet/GoogleLeNet trained with real ImageNet images and tested on BigGAN generated ImageNet images show classification accuracy 43/40/39/31, but STIC shows an accuracy: 63/63/83/80.

5. Discussion and Analysis

Discussion of Quantitative Results: From Table 1, we note that INN, WINN do not perform well due to training with ERM and learning from a weaker classifier. The PnP performance drops due to the apparent complexity while training the prior network. The STIC methodology supports the primary claim of a deep generative model of benefiting downstream tasks, such as classification. We hence see that the classifier in STIC methodology learns a tighter decision boundary (see improved **Cls**$_R$ and **Cls**$_G$) and smooth class interpolation to achieve this objective. However, the FID calculates the distance between feature vectors of real and generated images. We note that the classifier in STIC methodology learns a tighter decision boundary may not learn a good feature similarity of real and fake images, hence a slight drop in FID score w.r.t BigGAN. For classifier networks, we note a performance boost w.r.t SOTA classifier networks, thus showing the efficacy of our methodology as a classifier.

Generalizability of STIC Model: To understand the generalizability of the STIC method we adopt the precision-recall and k-nearest neighbor (KNN) analysis proposed in [19]. Fig 5 (a) shows high precision and recall at different initilizations, thus supporting our claim of diversity and generalizability in Sec 4. Similarly, we show precision and recall at different KNN using features of ResNet-50.

Ablation of Gram Matrices: In this work, we use the style representation of deeper layers, 'conv21'-'conv28' of STIC model and got FID: 15.01 on ImageNet. To show the effectiveness of style transfer from the deeper layers we do style transfer from shallow layers 'conv1'-'conv20' and that gives FID: 28 on ImageNe, thus not capturing more style. Howvever, considering all layers 'conv1'-'conv28' FID:30, mixes learning of deeper style and shallow layer style, thus leading to bad FID.

Running Time Complexity: Training our model was $\sim 3.2\times$ faster than training BigGAN and SNGAN. This is primarily because of the time taken for stabilization of GANs during training. Similarly, [25] optimizes two separate networks making their training time significantly larger. Also, INN [15] and WINN [21] trains multiple classifiers in a sequence ($>$ 25 number of classifiers in a sequence) for a single image synthesis, making its overall synthesis costly.

Effect of STIC on Other SOTA Classifiers: We ask ourselves whether a recurrent self-analysis method improves the classification accuracy of any classifier? We answer this

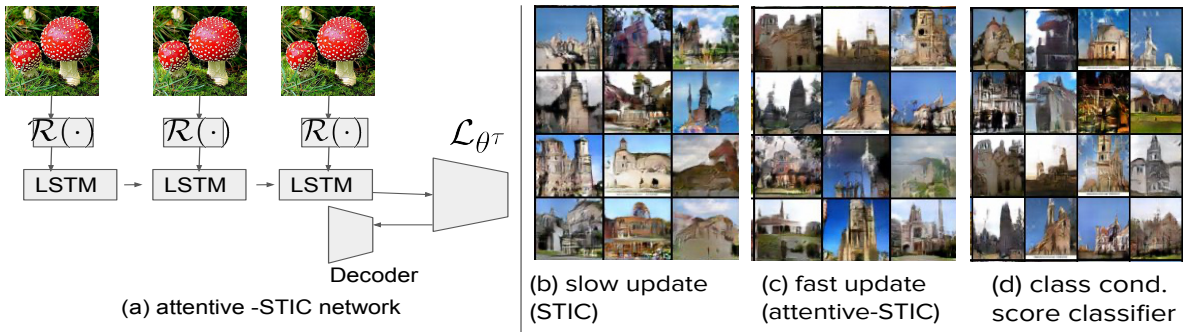

Figure 6: *(a-c)* ***attentive-STIC****:* STIC method can work in feature space. We show qualitative results of STIC and attentive-STIC on LSUN church at $10k$ iteration and note improved results. *(d)* ***score-STIC****:* we show the qualitative results of score-STIC only after $10k$ iterations. We show geometric details on these LSUN church samples.

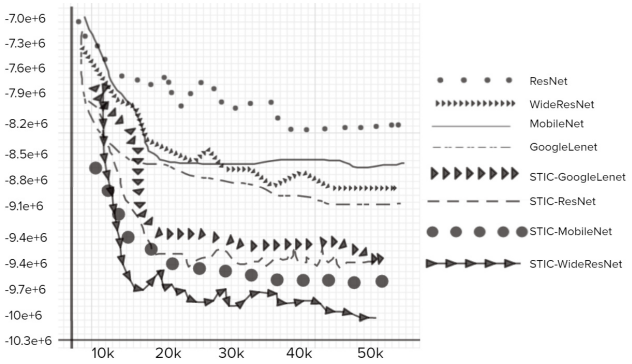

Figure 7: Loss (vertical axis) *vs.* No. of Iterations (horizontal axis) of discriminative classifier methods, STIC outperforms all.

in Fig 7. We show loss per iteration and STIC methodology improves the training accuracy of any classifier.

Optimal Number of Passes: In Sec 4 we show results for $\tau = 10$ number of passes. In this section, we study the number of passes and their relation with FID and other scores. We found that beyond $\tau = 10$ number of passes the synthesized image quality the FID and MIS scores minimally improves. Improving FID and MIS scores could be a possible future direction.

Attentive-STIC to Mitigate Slow Update of GRMALA: We tried to resolve the slow mixing of MALA-approx by trying GRMALA in the feture space instead of the pixel space. We adopt an attention based feature encoder [8] comprised of: (1) a reading network, $\mathcal{R}(\cdot)$ that receives an image $\mathbf{x}$ and decides to focus on a part of $\mathbf{x}$ using an attention mechanism (described later); (2) the $\mathcal{R}(\cdot)$ then outputs a vector $\mathbf{v}_t$ (which is rasterized from the patch being attended to); (3) an LSTM network receives $\mathbf{v}_t$ and provides a feature vector f. Similar to the DRAW [8] reading mechanism: $\hat{x}_t = x - \zeta(\hat{x}_{t-1}), v_t = \mathcal{R}(x, \hat{x}_t, v_{t-1}); [f_t, h_t^{enc}] = LSTM(v_t, h_{t-1}^{enc})$, here, $\zeta(\cdot)$ is a sigmoid function. The classifier, $p(y = y_c|f)$, now operates on the extracted feature of an image x and synthesize feature vector. The synthesize vector is passed to decoder network (see Fig 6) to upsample the feature vector to get synthesized image. The decoder is the DCGAN network. We show the qualitative results on LSUN church classes after one pass $\tau = 1$ (i.e. $5k$ iterations), see network in Fig 6(b) for results. In addition to that, the quantitative results are shown in Table 1.

Score-STIC a Class Conditional Score Discriminative Classifier: Based on our understanding from Eq 2, the STIC method depends on discriminative classifier. To this end, we propose a small modification on Wide ResNet architecture (or modification to any classifier network in general). The [29] method attempts to match the derivative of the model's marginal density with the derivative of the marginal density of real data using a score of a probability density $p(x)$, i.e. $\nabla_x \log p(x)$. We extended this idea and propose a novel class conditional score based Wide ResNet that we refer score-STIC. The WideResNet-28-10 last layer dimension is matched with input layer dimension (which is a criteria for score network [29]) followed by softmax classification. The following equation acts as a regularizer to the Eq 2, i.e.: $\mathbb{E}_{p_{data}(x)}\big[\frac{1}{2}||p_{\theta^\tau}(x)||_2^2 + tr(\nabla_x p_{\theta^\tau}(x)) + \frac{1}{2}||(y_c, p_{\theta^\tau}(y|x))||_2^2\big]$. We show results in Fig 6 and Table 1.

6. Conclusion

In this work, we emphasize on the relation $p(x|y) \propto p(y|x)$ and propose the STIC method to synthesize images using Gram-matrix Regularized MALA (GRMALA) sampler w.r.t class logit. Our classifier satisfies: (1) smooth interpolation; and (2) tighter class boundaries so as to generate photo-realistic samples. To this end, we propose a novel recurrent self-analytic STIC trained with VRM. We further show an Attentive-STIC model to address the slow mixing problem of GRMALA. In addition to that, we show a novel class conditional score function based Wide ResNet classifier and show improved generation on ImageNet, LSUN and Cifar10.

References

[1] J. R. Anderson. *Is human cognition adaptive?* na, 1991. 2

[2] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR '19)*, 2019. 5

[3] R. L. Campbell and M. H. Bickhard. If human cognition is adaptive, can human knowledge consist of encodings? *Behavioral and Brain Sciences*, 14(3):488–489, 1991. 2

[4] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik. Vicinal risk minimization. *Advances in Neural Information Processing Systems (NeurIPS '00)*, 13:416–422, 2000. 2, 4

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, 2009. 1, 5

[6] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pages 2414–2423, 2016. 2, 3, 4

[7] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations (ICLR '19)*, 2019. 1, 2, 3, 5

[8] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *Proceedings of the International Conference on Machine Learning (ICML '15)*, pages 1462–1471, 2015. 8

[9] S. Gurumurthy, R. K. Sarvadevabhatla, and R. V. Babu. Deligan: Generative adversarial networks for diverse and limited data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*, pages 166–174, 2017. 6

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pages 770–778, 2016. 5

[11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS '17)*, pages 6626–6637, 2017. 6

[12] G. Hinton, S. Osindero, M. Welling, and Y.-W. Teh. Unsupervised discovery of nonlinear structure using contrastive backpropagation. *Cognitive science*, 30(4):725–731, 2006. 3

[13] G. E. Hinton. Products of experts. 1999. 4

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 5

[15] L. Jin, J. Lazarow, and Z. Tu. Introspective classification with convolutional nets. In *Advances in Neural Information Processing Systems (NeurIPS '17)*, pages 823–833, 2017. 1, 2, 3, 5, 6, 7

[16] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '19)*, pages 4401–4410, 2019. 3

[17] D. Kotovenko, A. Sanakoyeu, S. Lang, and B. Ommer. Content and style disentanglement for artistic style transfer. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV '19)*, pages 4422–4431, 2019. 3

[18] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009. 1, 5

[19] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila. Improved precision and recall metric for assessing generative models. 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. 7

[20] J. Lazarow, L. Jin, and Z. Tu. Introspective neural networks for generative modeling. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV '17)*, pages 2774–2783, 2017. 1, 2, 3, 5

[21] K. Lee, W. Xu, F. Fan, and Z. Tu. Wasserstein introspective neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*, pages 3702–3711, 2018. 3, 7

[22] Y. Liu, Y. Zhou, X. Liu, F. Dong, C. Wang, and Z. Wang. Wasserstein gan-based small-sample augmentation for new-generation artificial intelligence: A case study of cancer-staging data in biology. *Engineering*, 5(1):156–163, 2019. 2, 5

[23] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953. 3

[24] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR '18)*, 2018. 5

[25] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*, pages 4467–4477, 2017. 3, 4, 5, 7

[26] A. Pal and V. N. Balasubramanian. Adversarial data programming: Using gans to relax the bottleneck of curated labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*, pages 1556–1565, 2018. 1

[27] G. O. Roberts and J. S. Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998. 3, 4

[28] G. O. Roberts, R. L. Tweedie, et al. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996. 3

[29] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS '19)*, pages 11918–11930, 2019. 3, 8

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pages 1–9, 2015. 5

[31] V. Vapnik and V. Vapnik. Statistical learning theory wiley. *New York*, 1:624, 1998. 2, 3

[32] M. Welling, R. Zemel, and G. E. Hinton. Self supervised boosting. *Advances in Neural Information Processing Systems (NeurIPS '02)*, 15:681–688, 2002. 3

[33] J. Xie, Y. Lu, S.-C. Zhu, and Y. Wu. A theory of generative convnet. In *International Conference on Machine Learning (ICML '16)*, pages 2635–2644, 2016. 5

[34] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 1, 5

[35] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[36] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR '18)*, 2018. 1, 2, 4

[37] Z. Zhang, Z. Wang, Z. Lin, and H. Qi. Image super-resolution by neural texture transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '19)*, pages 7982–7991, 2019. 3