Supplementary Section

In this section, we include more details which could not be included in the main paper due to space constraints.

- **Architecture Details and Algorithm:** discussion, analysis on the architecture, and pseudo code of the STIC methodology are provided in Sec 7.
- **STIC vs. SOTA Image Synthesis using Discriminative Classifier Methods:** we provide more analysis and justification on why and how the STIC learns class boundaries better than the baseline methods in Sec 8.
- **More Qualitative Results:** We provide more qualitative results in Sec 9.

7. Additional Architecture Details and Algorithm

In addition to the methodology described in Sec 3, we present the overall methodology in the form of an algorithm in 1.

Further Discussion on STIC: In continuation to the discussion in Sec 4, we now provide more details of the experiment for STIC. The initial learning rate is set to 0.0001 while we have a learning rate decay of 0.3 after every $10k$ epochs. Varying ϵ_1 between $\{0.9 - 1.0\}$ and ϵ_2 between $\{0.9 - 1.0\}$ to the range produces good quality images with high FID (see Fig 8) potentially because the learning from stochastic gradient langevin dynamics at passes $\tau \in \{1, 2, \cdots, T\}$ and learning from Gram matrix similarity of real images to the synthetic images support each other's learning. However, varying ϵ_3 between $\{0.01 - 0.02\}$ (i.e. lower range) helps to search the manifold and controls the diversity of synthesis. But, varying ϵ_3 between $\{0.1 - 0.8\}$ (i.e. higher range) does not provide a good update signal.

We choose WideResNet-28-10 due to various reasons, such as (1) we can directly drop the batch normalization without compromising the accuracy of the classifier; and (2) the architecture has less number of parameters, easy to train and widely used that has all the facilities of a standard ResNet architecture.

Further Discussion on Attentive stic: Read $\mathcal{R}(\cdot)$ and LSTM network of STIC are followed from [8]. The LSTM network provides a 100 dimensional feature vector to the discriminative classifier. The decoder network is the DCGAN, i.e. f100-conv1(8x8x1024)-conv2(64x64x512)-conv3(212x212x128)-conv4(512x512x3). The discrimnative classifier has the same hyperparamters as STIC.

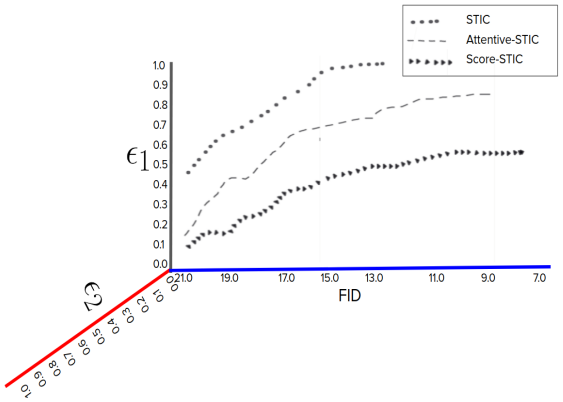

Figure 8: ϵ_1 ***vs.*** ϵ_2 ***vs.*** ***FID:*** we observe the highest FID score (blue axis) if the scaling factors ϵ_1 (black axis) and ϵ_2 (red axis) are set to the range $\{0.9 - 1.0\}$ while keeping the range of other scaling factor in the range $\{0.01 - 0.02\}$.

8. STIC vs. SOTA Image Synthesis using Discriminative Classifier Methods:

Our broad idea of using a discriminative classifier to synthesize class conditioned images may find similarity with earlier efforts [15, 20, 7], however, in many aspects, they are different from our STIC methodology:

- **STIC *vs.* INN:** The INN methodology [15] has $c \in \{1, 2, \cdots, C\}$ number of distinct CNN classifiers trained with ERM [31] and cascaded in a sequential manner. For any classifier, let's assume the c^{th} classifier, the parameters are $W_c = \{\mathbf{w}_c^0, \mathbf{w}_c^{1_1}, \cdots, \mathbf{w}_c^{1_K}\}$. The $\mathbf{w}_c^1$ denotes the weights of the top K separate layers for K classes, while $\mathbf{w}_c^0$ carries all internal features. The negative samples are sampled for each class. Such negative samples along with the real samples are then utilized by the c+1^{th} classifier to segregate real samples to negative samples. On the other hand, the STIC classifier serves dual objectives, *viz.* the interpolated samples from one class to another must be smooth and the classifier must learn tighter class boundaries so as to generate photo-realistic samples. Thus, STIC is different from INN in several ways, such as: (1) STIC is trained with VRM [36] with virtual image-label pairs along with real image-label pairs that provide a good learning of smooth class boundaries and tighter class boundary across passes; (2) the loss function and the training methodology of STIC is different from INN, i.e. INN uses a separate branch for each classes, but, STIC instead uses a single architecture wide ResNet (ref. Sec 4) and trains the method; (3) STIC optimizes less number of parameters (single architecture) than INN (classifier with multiple branches), and, we note that, the convergence time and image quality of STIC is far better than INN due to training the classifier with

Algorithm 1: Training STIC

Input: Number of passes $\tau \in \{1, 2, \cdots, T\}$, Minibatch size: m
Output: Trained STIC model
for *a pass* τ **do**
  iteration = 0
  for *iteration* $\leq 5k$ **do**
    Draw a batch of K synthesized image-label pairs from previous classifier $p_{\theta^{\tau-1}}$ using GRMALA in Eq 3;
    Draw a batch of K synthesized *mixup* image-label pairs from previous classifier $p_{\theta^{\tau-1}}$ using GRMALA in Eq 3;
    Draw a batch of N image-label pairs from dataset p_{data};
    Draw a batch of N *mixup* virtual image-label pairs similar to [36];
    Update weight θ^{τ} of STIC classifier using mini-batch stochastic gradient descent with gradients as computed below:

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

$$-\sum_{(x_i,y_i)\sim p_{\theta^{\tau}}}^{i=1,\cdots,N} \log p_{\theta^{\tau+1}}(y_i = -1|x_i) - \sum_{(x_k^{mixup},y_k^{mixup})\sim p_{\theta^{\tau}}}^{k=1,\cdots,K} \log p_{\theta^{\tau+1}}(y_k = -1|x_k^{mixup})$$

  end
end

VRM; (4) utilization of synthetic samples as fake sample is different from INN; and most importantly (5) our sampling technique, i.e. GRMALA (ref Sec 3), is novel and different from the MCMC-based sampling of INN.

- **JEM vs STIC:** The JEM [7] methodology is developed based on an energy based estimation of $p(x)$ and $p(x, y)$. We note that such a method is different from ours, as: (1) as described above, the STIC uses VRM based training and GRMALA. (2) The recurrent class boundary re-estimation way of training is different from the JEM methodology.

We ask ourselves the question, *how does VRM help the STIC method?* From the understanding of VC theory [31], the classification error of a classifier $\hat{f}$ can be decomposed as:

$$R(\hat{f}) - R(f) \leq O\Big(\frac{|\hat{\mathcal{F}}|_C}{n^{\alpha}}\Big) + \epsilon \tag{6}$$

here, $f \in \mathcal{F}$ is the true classifier function we wish to approximate using the function $\hat{f} \in \hat{\mathcal{F}}$. The $|\cdot|_C$ is the class capacity measure, error is the R, number of data points are shown as n and α is the learning rate. We note that, the ϵ is the approximation error of $\hat{\mathcal{F}}$ with respect to the function $\mathcal{F}$. To this end, a loss function $l(\cdot)$ penalizes the difference between the predictions $\hat{f}(x)$ and the ground truth y sampled from $p_{data}(x, y)$. The average of the loss function $l(\cdot)$ is averaged over training data samples and the empirical risk is minimized as follows:

$$R(\hat{f}) = \sum_{x_i,y_i\in p_{data}(x,y)}^{i=1,\cdots,n} l(\hat{f}(x_i), y_i) \tag{7}$$

A classifier function $\hat{f}$ trained with STIC takes the following form:

$$\begin{aligned} R(\hat{f}) = & \sum_{(x_i,y_i)\sim p_{data}}^{i=1,\cdots,N} l(\hat{f}(x_i), y_i) \\ & + \sum_{(x_k^{mixup},y_k^{mixup})\sim p_{mixup}}^{k=1,\cdots,K} l(\hat{f}(x_k^{mixup}), y_k^{mixup}) \\ & + \sum_{(x_i,y_i)\sim p_{\theta^{\tau}}}^{i=1,\cdots,N} l(\hat{f}(x_i), -1) \\ & + \sum_{(x_k^{mixup},y_k^{mixup})\sim p_{\theta^{\tau}}}^{k=1,\cdots,K} l(\hat{f}(x_k^{mixup}), -1) \end{aligned} \tag{8}$$

Similar to the argument presented in [4], if the virtual image-labels are a poor approximation of class vicinity then STIC trained with VRM performs at least as good as a classifier trained with ERM. We note that the virtual image-labels using mixup of softmax [36] provides a good ap-

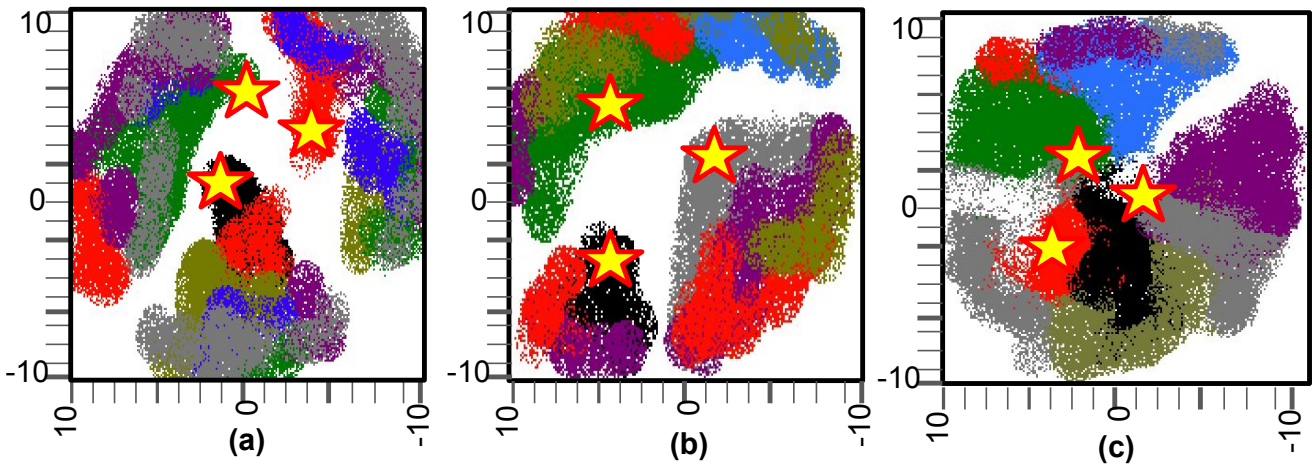

Figure 9: ***Visualization of CIFAR 10 dataset Class Boundaries***: We visualize seven class (i.e. airplane, automobile, bird, cat, deer, dog, frog classes of CIFAR 10 are shown to reduce the clutter) boundaries of (a) INN (b) JEM and (c) STIC on the CIFAR 10 dataset. We observe that the class boundary is very compact in STIC. The yellow stars are points we sample and synthesize images.

proximation of class vicinity. In addition to that, the recurrent self-estimation with VRM is a better approximation of class vicinity w.r.t the method proposed in [36]. We show the class boundary visualization of INN, JEM and STIC in Fig 9 and we note that the STIC class boundary is compact - supporting our claim. We also note that, the use of GRMALA based synthesis also provides good learning signal to estimate class boundaries.

9. More Synthesized Images

In addition to our qualitative results shown in Fig 1, in this section we show more qualitative images of LSUN, Cifar 10 and ImageNet datasets in Figs 11-15 (please see next pages).

10. Synthesizing using STIC

Following the training process described in Sec 3 and Algorithm 1, STIC synthesize images as follows: starting with an initial x_0 typically sampled from a Gaussian distribution $\mathcal{N}(0, I)$, the GRMALA uses the transition operator, *viz.* $x_{t+1} = x_t + \epsilon_1 \nabla \log p(x_t) + \sum (G^L(x_t) - A^L(x_t))^2 + \mathcal{N}(0, \epsilon_2^2)$, synthesize novel image samples from the classifier at pass $\tau = T$, see Fig 10 (b). We show the training process again in Fig 10 (a).

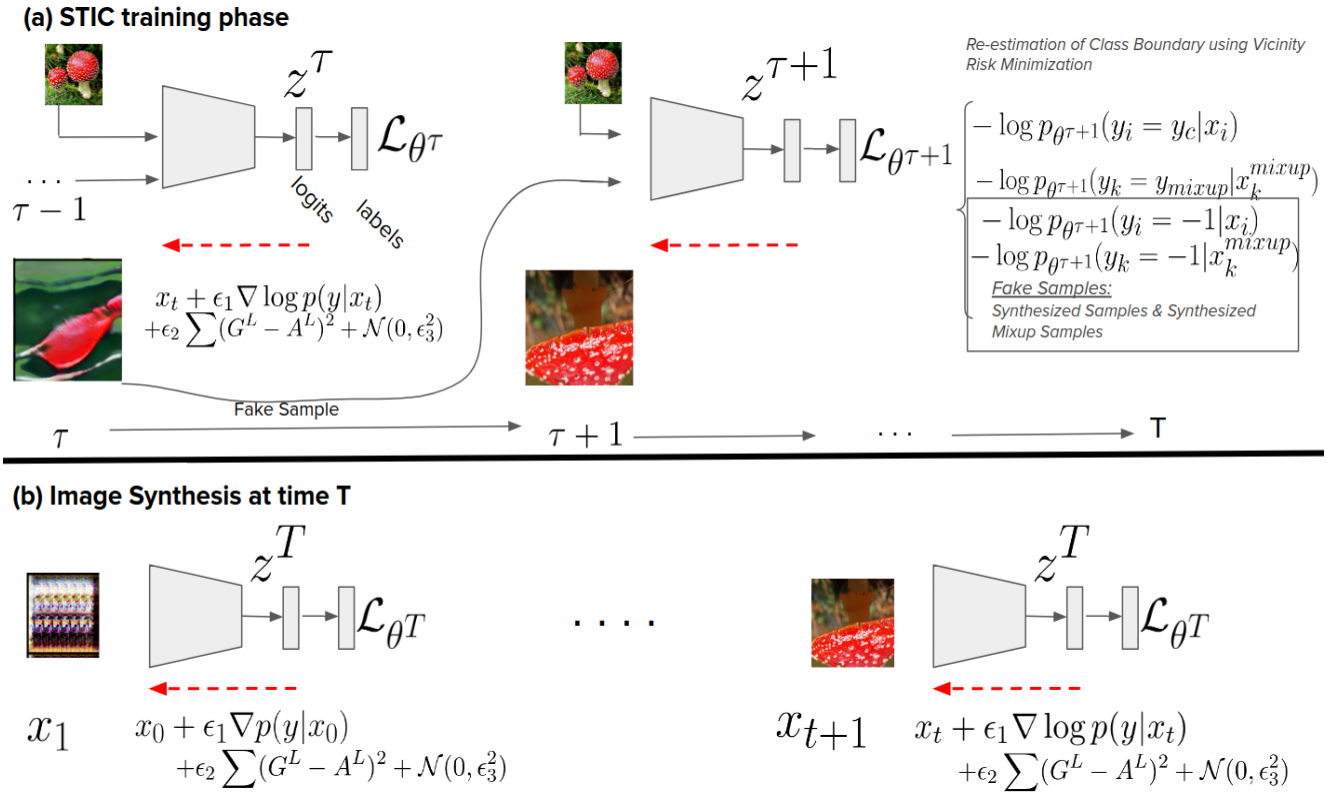

Figure 10: ***Image Synthesis using STIC at Image Generation Phase:*** (a) training phase of STIC, (b) image synthesis from STIC at time t.

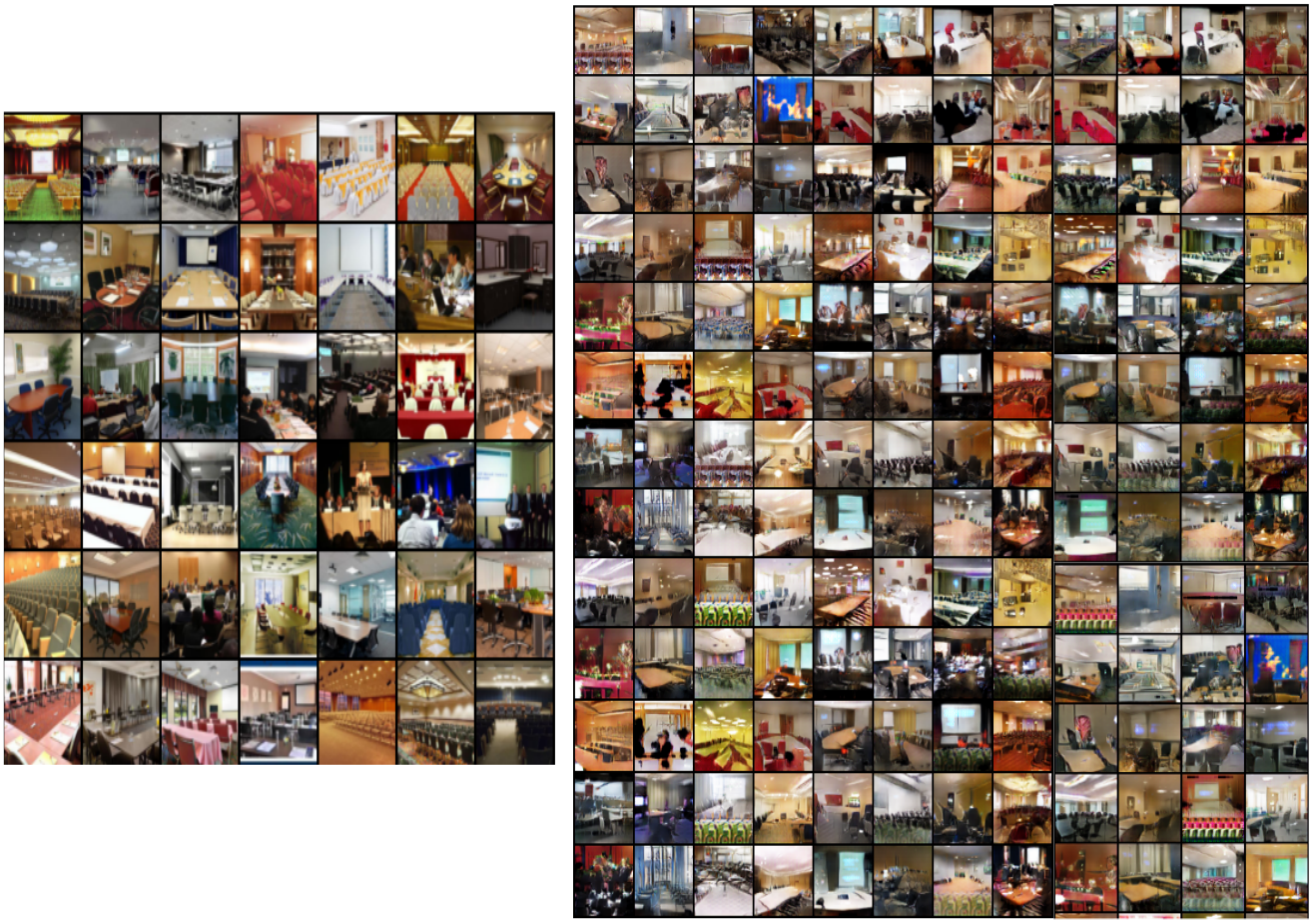

Figure 11: ***More Qualitative Results on the LSUN dataset:*** We show qualitative results on the LSUN conference class.

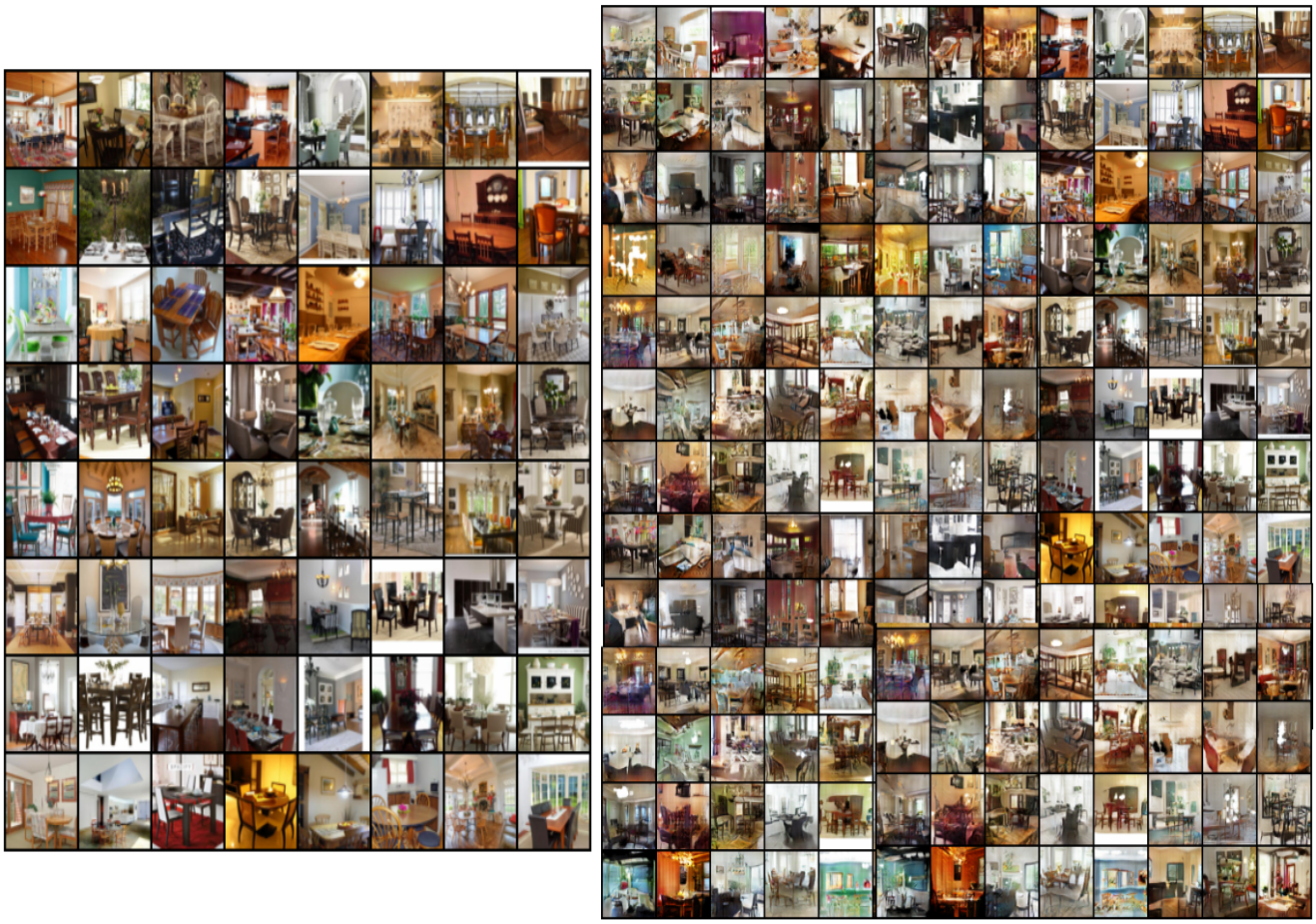

Figure 12: ***More Qualitative Results on LSUN dataset:*** We show qualitative results on LSUN dinning hall class.

Figure 13: ***More Qualitative Results on LSUN dataset:*** We show qualitative results on LSUN classroom class.

Figure 14: ***More Qualitative Results on CIFAR 10 Dataset:*** We show qualitative results on CIFAR 10 images (mixed classes).

Figure 15: ***More Qualitative Results on ImageNet Dataset:*** We show qualitative results on ImageNet images.

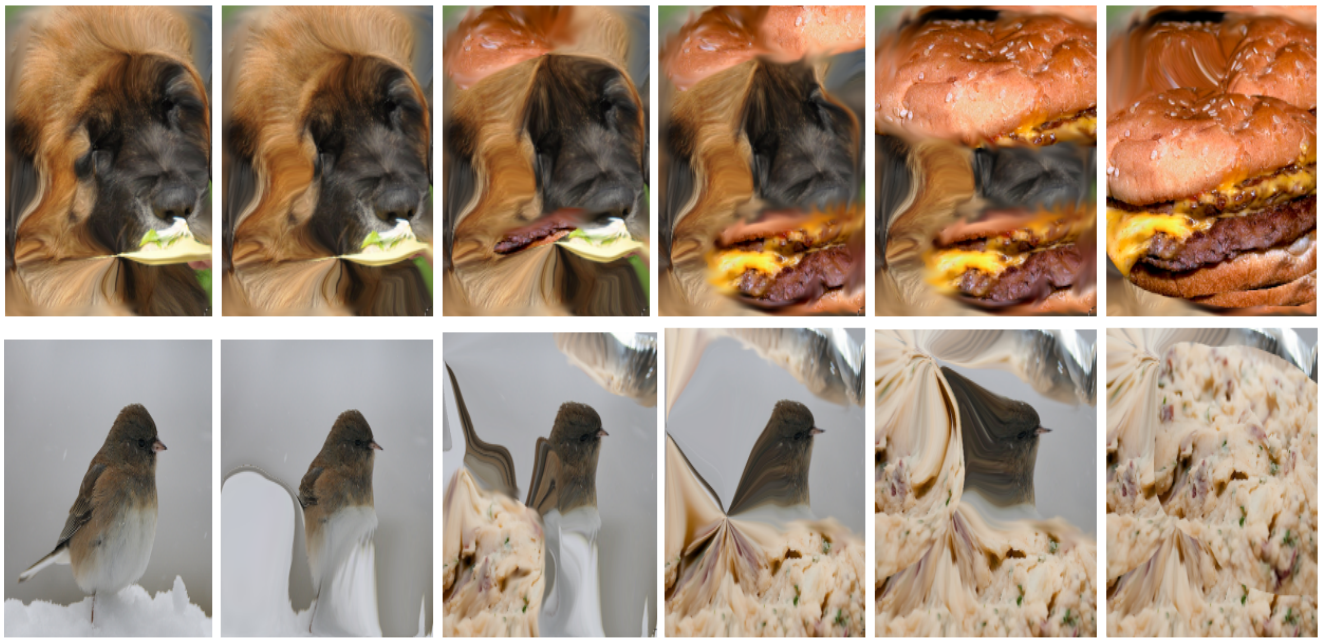

Figure 16: ***Class Interpolation Results on ImageNet Dataset:*** We show two more interpolation results on ImageNet images.