NaturalInversion: Data-Free Image Synthesis Improving Real-World Consistency

Yujin Kim[1,2], **Dogyun Park**[1,2], **Dohee Kim**[2], **Suhyun Kim**[2*]

[1]Korea University, Republic of Korea
[2]Korea Institute of Science and Technology, Republic of Korea
{lakeeye1220, koparkrea, kimdohe1070, dr.suhyun.kim}@gmail.com

Abstract

We introduce *NaturalInversion*, a novel model inversion-based method to synthesize images that agrees well with the original data distribution without using real data. In NaturalInversion, we propose: (1) a *Feature Transfer Pyramid* which uses enhanced image prior of the original data by combining the multi-scale feature maps extracted from the pre-trained classifier, (2) a *one-to-one* approach generative model where only one batch of images are synthesized by one generator to bring the non-linearity to optimization and to ease the overall optimizing process, (3) learnable *Adaptive Channel Scaling* parameters which are end-to-end trained to scale the output image channel to utilize the original image prior further. With our NaturalInversion, we synthesize images from classifiers trained on CIFAR-10/100 and show that our images are more consistent with original data distribution than prior works by visualization and additional analysis. Furthermore, our synthesized images outperform prior works on various applications such as knowledge distillation and pruning, demonstrating the effectiveness of our proposed method.

Introduction

Convolution Neural Networks (CNNs) have achieved a great success in various computer vision tasks such as image classification, image generation, etc (Simonyan and Zisserman 2014; Goodfellow et al. 2014). The emergence of large datasets (Krizhevsky and Hinton 2009; Deng et al. 2009) has led to a progressive improvement in various applications (Zhai et al. 2021; Chu et al. 2020). The CNNs learn meaningful feature spaces with rich information from low-level features to high-level semantic content (Zeiler and Fergus 2014). Hence, several works have been proposed to transfer knowledge from pre-trained networks into a lightweight model to inference in various conditions. Specially, Knowledge Distillation (Hinton, Vinyals, and Dean 2015) transfers the knowledge from the pre-trained teacher network to student networks. Network pruning (Liu et al. 2017) reduces redundant neural connections of a pre-trained network and fine-tunes the remaining weights to recover the accuracy. However, these methods require original data to train or fine-tune the compressed networks, limiting their use in privacy-sensitive or data-limited scenarios.

To overcome these scenarios, several approaches have synthesized the data from a pre-trained network without original datasets or images prior (Mahendran and Vedaldi 2015). Recently, Yin et al. (2020) has proposed a novel method to optimize the raw input space $\hat{x}$ to synthesize images, using a regularizer that follows the statistics in the batch-normalization(BN) layers of a pre-trained classifier. However, we argue that optimizing the high-dimensional raw input space is challenging for two reasons: 1) limited image prior since the statistics in BN layer are averaged, ignoring the specific information for individual images, 2) optimization without non-linearity. These problems lead to sub-optimized images which are heavily inconsistent with original data distribution: low fidelity and low diversity of synthesized images. Therefore, these images cause performance degradation in applications compared to performance trained with original dataset.

Therefore, for the richer image prior, we focus on features of pre-trained CNNs, since they encode the low to high level image prior on multi-scale feature maps (Islam, Jia, and Bruce 2020). The sparsity in feature maps eliminates the irrelevant variability of the input data while preserving the important information (Xu et al. 2021). Furthermore, feature maps of each layer encourage to capture multi-scale characteristics of the target class and real data distribution. Thus, utilizing the feature maps of the pre-trained classifier for generation strengthens the real data characteristics on the synthesized images (Shocher et al. 2020; Wang et al. 2021).

Inspired by the above methods, we propose the sub-generator: *Feature Transfer Pyramid* (FTP) that uses multi-scale feature maps from the pre-trained network to generated images. Thus, FTP enhances the characteristics of real data by sequentially combining the multi-scale feature maps from a pre-trained classifier. By adding the final combined multi-scale feature maps to synthesized image, it strengthens the original data characteristics on the synthesized images, making images more consistent to original data distribution. Secondly, we use the CNN-based generator to bring a non-linearity in optimization, which makes generator possess higher ability to represent the complex characteristics. We further propose the *one-to-one* approach, which utilizes a generator for one batch of images during the current batch

*Corresponding author.

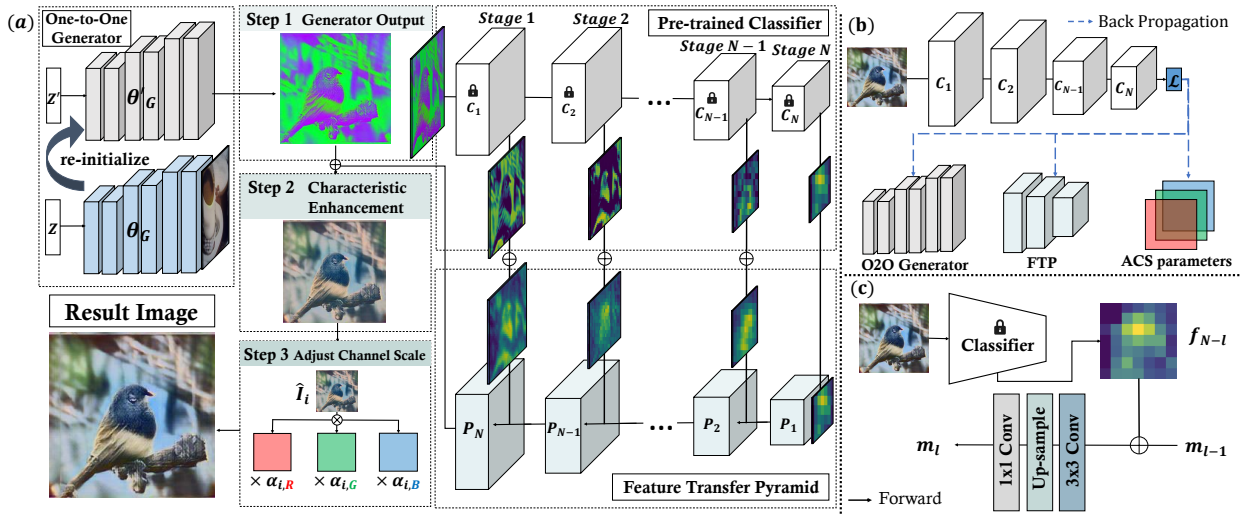

Figure 1: (a) Overall diagram of NaturalInversion for synthesizing high-quality images given pre-trained classifier. NaturalInversion pipeline consists of 3 parts : Step1. The generator produces one batch of samples and re-initializes after the current batch synthesis step. Step 2. Feature Transfer Pyramid (FTP) works in full conjunction with a pre-trained classifier, extracting feature maps and enhancing the characteristics of the original data. Step 3. Multiply the Adaptive Channel Scaling parameters to Step2 images, and then we obtain final images. (b) Backpropagation phase to feed final images to a pre-trained classifier and update all components. (c) Illustration of a single FTP block.

synthesis step. Specifically, each generator is optimized to generate a specific subset of the original datasets, which accelerate the convergence to the specific optimal point. In addition, we use the learnable *Adaptive Channel Scaling* (ACS) parameters that are end-to-end trained to adjust the output channel scale to find "optimal channel scale range" of original data learned by the classifier. Thus, ACS parameters implicitly learn the image prior.

In the end, with our proposed method called *NaturalInversion*, we synthesize more optimized images which agrees more with original data distribution: higher fidelity and diversity than prior works. Experiments on the CIFAR dataset show that NaturalInversion not only synthesize more visually plausible samples than prior works but also achieves significantly higher performance on empirical studies and applications in data-free conditions. The contributions of our proposed methods can be summarized as follows:

- We propose a *Feature Transfer Pyramid*, which enhances the synthesized images with the multi-scale feature maps extracted from a pre-trained classifier.
- We introduce the *one-to-one* approach based generator to ease the optimizing process and bring a non-linear optimization.
- We further implicitly learn the image prior of original data channel scale by end-to-end training the *Adaptive Channel Scaling* parameters.

Related Work

Model Inversion. Inverting a pre-trained model helps to interpret and understand the deep representation that the model stored. Several works have reconstructed what the model learned by optimizing input space from noise to image with regularizers. DeepDream (Mordvintsev, Olah, and Tyka 2015) visualizes the pattern that the model watch by selecting one or more layers of the model and optimizing the input space to maximize the class probability using the arbitrary target label $\hat{y}$. Follow-up studies such as DAFL (Chen et al. 2019) simultaneously perform inversion and knowledge distillation, training a generator to minimize the cross-entropy loss and maximize the representation of the pre-trained teacher network. DFAD (Fang et al. 2019) trains the generator that maximizes disagreement between the predictions of the pre-trained teacher and the student model, while a student network reduces the discrepancy with a teacher to generate more confusing samples. However, DAFL and DFAD concentrate on synthesizing the images for a specific task, such as knowledge distillation. Thus, these images are far from the original data distribution, leading to performance degradation on various vision applications. To alleviate this issue, Yin et al. (2020); Haroush et al. (2020) use a regularizer for channel-wise feature distribution matching with stored BN statistics to improve consistency with original data distribution, achieving the improvement in other applications. However, due to the problems with optimizing the input space with limited image prior, their images still lack of fidelity and diversity.

Usage of Feature Maps. The feature maps from pre-trained CNNs can provide insight into which characteristic in feature map does the classifier intensively detects for. The low-level feature maps generated by the shallower layers of CNNs encode basic representations such as edges and corners of real images. The deeper features include more complex semantic representations such as complicated geo-

metric shapes in their feature maps (Yosinski et al. 2015). For confirming the encoded information of CNNs, many works visualize feature maps of a pre-trained model to interpret feature representation in intermediate layers (Zeiler and Fergus 2014; Olah, Mordvintsev, and Schubert 2017). Inspired by previous works, several studies have synthesized realistic images using the properties of feature maps (Kalischek, Wegner, and Schindler 2021; Heitz et al. 2021; Xu et al. 2021). Many style transfer algorithms (Gatys, Ecker, and Bethge 2016; Lin et al. 2021; Kalischek, Wegner, and Schindler 2021) extract the multi-scale feature maps on multiple layers by forwarding the style image to the network. Then they render the style from an image to a content image while preserving the semantic information of the content image. Gatys, Ecker, and Bethge (2015) utilizes the correlations between feature maps in several layers to capture spatial-based texture representations. GAN-based Semantic Generation Pyramid (Shocher et al. 2020) proposes to feed the target image to a pre-trained classifier and combine both the feature map and same level generator block output to replicate the classifier features. We draw inspiration from several tasks of using multi-scale feature maps to obtain different scale representations of pre-trained CNNs. Our proposed method gradually enhances the real data characteristics encapsulated in a pre-trained teacher network to capture the distribution of the original dataset.

Method

We aim to synthesize more realistic images which agrees well with the original data distribution. In this section, we provide a brief background and introduce our *NaturalInversion* methods.

Preliminary

Model inversion approaches update the input space to get an image $\hat{x}$ given the pre-trained teacher network T. Given trainable input space $\hat{x}$ with randomly initialized noise and arbitrary target label y, the input space is updated by minimizing the below objective function.

$$\min_{\hat{x}} \mathcal{L}(\hat{x}, y) + \mathcal{R}(\hat{x}) \tag{1}$$

We can divide objective function into two part, loss function $\mathcal{L}$ and regularizer $\mathcal{R}$ that captures natural image prior.

Inception Loss. Inceptionism-style images synthesis approach (Mordvintsev, Olah, and Tyka 2015) uses inception loss to maximizes the probability of the expected target class produced by the pre-trained teacher model. Given an arbitrary target label $\hat{y}$, it encourages minimizing the cross-entropy loss to find out the class probability distribution of original data, as below:

$$\mathcal{L}_{CE} = CE(T(\hat{x}), y) = -\sum_{i} \hat{y}_i \log y_i \tag{2}$$

Feature Distribution Regularizer. For more image prior, DI (Yin et al. 2020) synthesizes the images that follow the original dataset distribution stored in the pre-trained teacher network. Given running mean ($\hat{\mu}$), and running variance($\hat{\sigma}^2$) of each BN layer (Ioffe and Szegedy 2015), they minimize the difference between the channel-wise mean($\mu(\hat{x})$) and variance($\sigma^2(\hat{x})$) of synthesized images and running statistics of every BN layer.

$$\mathcal{R}_{BN} = \sum_{i=1}^{l} (\|\mu_i(\hat{x}) - \hat{\mu}_i\|_2 + \|\sigma_i^2(\hat{x}) - \hat{\sigma}_i^2\|_2) \tag{3}$$

where $\mu_i(\hat{x})$ and $\sigma_i^2(\hat{x})$ are the mean and variance of output feature maps from each network layer i, and $\hat{\mu}_i$ and $\hat{\sigma}_i^2$ denote the running mean and variance of pre-trained network.

Image Prior Regularizer. DeepDream(Mordvintsev, Olah, and Tyka 2015) uses image prior regularizers forcing synthetic images to be stably optimized following pre-defined prior:

$$\mathcal{R}_{prior} = \lambda_{TV}\mathcal{R}_{\mathcal{TV}}(\hat{x}) + \lambda_{l_2}\mathcal{R}_{l2}(\hat{x}) \tag{4}$$

$\mathcal{R}_{\mathcal{TV}}$ penalizes the sparsity in synthesized images with scaling factor λ_{TV}. l_2 normalization, $\mathcal{R}_{l2}(\hat{x}) = \|\hat{x}\|_2$, encourages the synthesized images to have a small norm with scaling factor λ_{l_2}.

Inversion Using Generative Model

Our goal is to encourage the generator to implicitly approximate the original dataset distribution $p_{real}(x)$. We use the conventional conditional GAN (Mirza and Osindero 2014) concept: the objective of the generator is to map z from $p_z(z)$ to original data space $\mathcal{X}$ from $p_{real}(x|y)$. We sample the latent vector z from the normal distribution $\mathcal{N}(0, 1)$ concatenated with y encoded as one-hot vector. However, if the latent vector changes every training epoch, the latent vectors are largely ignored or have minor effects on the variations of the images in our framework. To address the mode-collapse problem, we propose a "one-to-one" approach, which utilizes one generator for synthesizing one batch of samples during the current batch synthesis step B. Then, after sufficiently optimizing θ_G for a latent vector z, we re-sample z' and re-initialize the generator weight from θ_G to θ'_G. As a result, the continuously re-initialized generator specifically captures a unique sample x corresponding to each z by optimizing the θ_G. Therefore, our generator can synthesize images from various modes with respect to the different B, leading to diverse images. The generator architecture are shown in the appendix.

Feature Transfer Pyramid

Our goal is to utilize the richer image prior from the pre-trained classifier. Thus, we propose *Feature Transfer Pyramid*(FTP) as a sub-generator, a novel hierarchical schema to gradually capture the distribution of the original dataset from different scale feature maps. As shown in Fig.1, FTP operates in conjunction with a pre-trained classifier. More specifically, given the generator output $G(z|y)$, we feed it into the pre-trained classifier and extract the feature maps F=$\{f_1, f_2, f_3, ...f_N\}$ with different layers. The f_1 denotes the low-level feature maps, and f_N is higher-level feature map. We extract the feature maps at the downsampling point

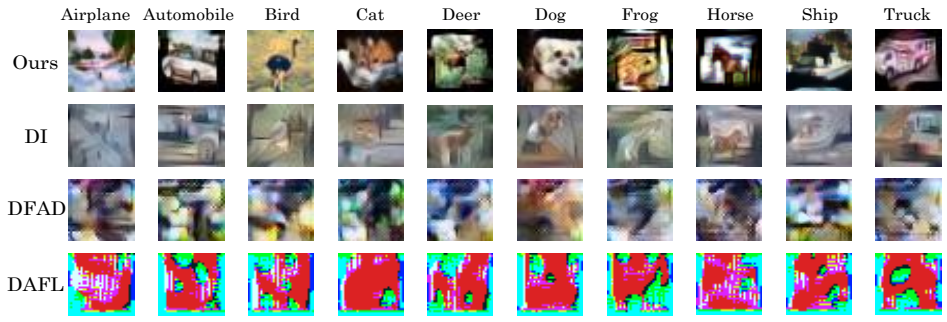

Figure 2: Visualization of CIFAR-10 examples compare to prior works. From bottom to up, the result images were synthesized with DAFL, DFAD, DI, and our method, respectively. All images are synthesized by the same pre-trained classifier, ResNet-34. The CIFAR-10 class labels are written at the top.

because different scale feature maps have different representations of a pre-trained classifier. The low-level feature maps include the simple and primary characteristics, and the higher-level feature map increasingly represents complex semantic representation. Fig.1(c) shows the building block of FTP. Now, we can get the outputs of each block, the Feature enhancement map M=$\{m_1, m_2, ...m_L\}$, as:

$$m_l = \begin{cases} W_1^l(\Phi(f_N)), & \text{if } l = 1 \\ W_1^l(\Phi(W_3^l(m_{l-1} \oplus f_{N-l}))), & \text{otherwise} \end{cases} \quad (5)$$

where $\Phi(\cdot)$ denotes the upsample layer, W_1^l is the l_{th} 1×1 convolution layer, and W_3^l denotes the l_{th} 3×3 convolution layer of FTP. First, f_N, the last feature map, is upsampled to fit the first FTP output size and fed to FTP as input. Then, the output of FTP block is summed with the f_{N-1} which is a prior stage feature map from a classifier. Each output of FTP block is gradually added to feature maps at the corresponding stage of the classifier and undergoes convolution operations except the first block. Finally, the characteristic of original data is enhanced in the synthesized images by adding the m_L to $G(z|y)$ element-wisely. FTP gradually captures the different spatial characteristics in the target class, and it leads to synthesized images which resembles the real data. We build up the layers of FTP when the spatial resolution change. More detailed structure of FTP are in the appendix.

Adaptive Channel Scaling

The pre-trained classifier is trained with dataset which is scaled to [0,1] and normalized with its mean and variation. Thus, the pre-trained classifier is biased to channel scale range of normalized original data, which can be used as image prior. To implicitly utilize the optimal scale range of original data, we propose the *Adaptive Channel Scaling* parameters α, which are learnable.

$$\alpha \in \mathbb{R}^{B \times C \times 1 \times 1} \quad (6)$$

Where the B denotes the batch size, and the value of C is 3 known as RGB channel. We iteratively multiply α to final images during inversion steps to adjust the output channel distribution of synthesized images. Finally, we can get the final synthesized images $\hat{I}$ using below equation:

$$\hat{I}_i = \alpha_i \otimes (m_L \oplus G(z|y)) \quad (7)$$

Figure 3: One random batch samples of CIFAR-10 images. Our proposed method can synthesize the diverse images within one batch. Each column represents the same target class image. The objects in mini-batch images has different shapes and colors.

where $\oplus$ and $\otimes$ means element-wise add/multiply individually, and m_L is the last Feature enhancement map in Eq. 5. We multiply α to combination map with generator output and m_L. This ACS parameters induce the synthesized images to the "optimal scale range" which leads to produce the suitable loss for synthesizing original distribution. Finally, we again feed these final images to the pre-trained classifier T and produce the gradient. Our generator, FTP and α are optimized by same objective function which is defined identically as Eq. 1.

$$\mathcal{L}_{inv}(\hat{I}, y) = \mathcal{L}_{CE}(\hat{I}, y) + \mathcal{R}_{BN}(\hat{I}) + \mathcal{R}_{prior}(\hat{I}) \quad (8)$$

After finishing the inversion epochs, the generator, FTP and α are re-initialized and synthesize the next batch of images. The overall process of NaturalInversion is summarized in Alg. 1.

Experiments

In this section, we evaluate the performance of NaturalInversion on CIFAR-10 and CIFAR-100 (Krizhevsky and Hinton 2009). Our experiments contain two parts, (1) analysis of our method: we verify that our method can synthesize more natural images (2) applications: we ensure the effectiveness of NaturalInversion on various applications in data-free conditions. Our experiments settings can be found in appendix.

Algorithm 1: NaturalInversion Algorithm

Require: A pre-trained teacher network T
Output: Inversion Images $\hat{I}$
for number of batches B **do**
  initialize Generator $G(\cdot;\theta_G)$, FTP $P(\cdot;\theta_P)$, α
  $z \leftarrow \mathcal{N}(0,1)$
  for inversion epoch E **do**
    $\hat{x} \leftarrow G(z|y)$ ▷ generator output
    $m_L \leftarrow P(\hat{x})$ ▷ FTP output
    $\hat{x}' \leftarrow m_L + \hat{x}$ ▷ add last FTP output to $\hat{x}$
    $\hat{I} \leftarrow \alpha \times \hat{x}'$ ▷ synthesize the final image
    $\mathcal{L}_{inv} \leftarrow T(\hat{I})$ by Eq.8 ▷ compute loss function
    $\theta_G \leftarrow \theta_G - \eta_G \nabla \theta_G \mathcal{L}_{inv}$
    $\theta_P \leftarrow \theta_P - \eta_P \nabla \theta_P \mathcal{L}_{inv}$
    $\alpha \leftarrow \alpha - \eta_\alpha \nabla \alpha \mathcal{L}_{inv}$
  end for
  return Mini batch Images $\hat{I}$
end for

Analysis of NaturalInversion

We perform several studies to verify that synthesized images using our methods capture the original dataset distribution. As part of our experiments, we compare the (a) visualization, (b) t-SNE, (c) comparison of generative model evaluation metric result and (d) from-scratch training experiments.

Visualization. We show that our synthesized images of CIFAR-10 are more visually realistic than prior works in Fig. 2. This is because FTP further captures the characteristics of the target class: the shape or color of individual samples are more clearly reflected in synthesized images. Furthermore, we show that our "one-to-one" approach synthesizes more diverse images by optimizing particular generator for particular subset of original data as shown in Fig. 3. In conclusion, our method are more consistent with original data distribution with high fidelity and diversity than prior works.

Feature Visualization. To verify that our method encourages capturing original data distribution, we visualize the feature space from each residual block output and embedding layer of ResNet-34 using t-SNE (Van der Maaten and Hinton 2008). For visualizing feature space, we use the ResNet-34 trained by CIFAR-10 and compare the low-level feature space among 10k original CIFAR-10, DI, and our method. Entire t-SNE results are in the appendix. As shown in Fig. 4, the feature representations of DI have low diversity and are heavily different from the original distribution. In contrast, the feature representation of our method is more similar to original data than DI, which verifies that our method fairly estimates the internal feature representation of real data.

Generative Model Evaluation Metric. We further analyze our methods with the generative model evaluation metric to assess how similar our images are to the original dataset : (a) single-value metric - Inception Score (IS) (Salimans et al. 2016) and Frechet Inception Distance (FID)

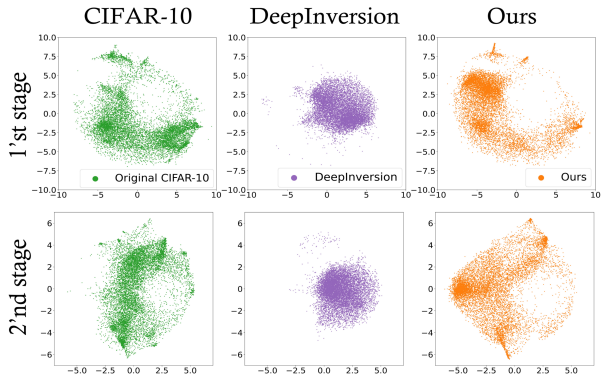

Figure 4: t-SNE of original CIFAR-10, DI, and ours. We extract the feature embedding from 1st and 2nd residual blocks. The green denotes the randomly sampling 10k original CIFAR-10, The orange is NaturalInversion 10k images and DI 10k images denote purple color.

Dataset	Method	IS	FID	Precision	Recall
CIFAR-10	WGAN-GP	7.86	29.3	0.7040	0.4353
	DI	2.67	197.33	0.5996	0.0028
	Ours	5.15	76.04	0.6797	0.2792
CIFAR-100	WGAN-GP	6.68	37.30	0.7007	0.4023
	DI	4.04	151.52	0.4051	0.0119
	Ours	6.40	62.90	0.6845	0.2894

Table 1: Metric result of synthesized images by WGAN-GP, DI, and ours. A higher score of IS, Precision and Recall is better whereas a lower score of FID is better.

(Heusel et al. 2017). (b) two-value metrics- Precision and Recall (P&R) (Sajjadi et al. 2018). We specifically verify the qualitative fidelity and diversity of images by the two-value metrics. We synthesize 50k CIFAR-10/100 images from ResNet-34 and use the logit and embedding layer of ImageNet pre-trained Inception-v3 (Szegedy et al. 2016) for calculating the IS and remaining metrics respectively. To show how well we approximated the original data distribution, we compare the synthesized images with the images by DI and GAN-based model, WGAN-GP, which utilizes the original data. As shown in Table 1, NaturalInversion outperforms DI across CIFAR-10/100 in terms of IS, FID, P&R. In addition, NaturalInversion is even close to WGAN-GP baselines without original data. Through these evaluation metrics, we confirm that our method estimates the original distribution well without the real data.

From Scratch Training. To demonstrate how well our method captures the original distribution, we train the model from scratch with images by our method without any real data and compare the training accuracy of model with other methods: DAFL and DI. First, we synthesize 256k images using DAFL, DI, and our methods from ResNet-34, VGG-16 trained by CIFAR-100 dataset. Then, we train the randomly initialized networks from scratch using synthesized images. We set the mini-batch size as 128 and train the model for 200 epochs using an SGD optimizer with a 0.05 learning

Inversion Model	Train Model	DAFL	DI	Ours
ResNet-34	ResNet-18	32.40	49.61	**74.55**
	VGG-11	15.85	39.46	**67.31**
	MobileNetV2	33.49	34.86	**64.34**
VGG-16	ResNet-18	1.96	46.90	**72.38**
	VGG-11	1.90	41.94	**67.95**
	MobileNetV2	2.11	31.32	**62.86**

Table 2: From scratch experiments of CIFAR-100. We train the various models from scratch using synthesized images without any information about the training set or assistance from other pre-trained networks.

O2O	FTP	ACS	K.D acc	IS	FID	Precision	Recall
DeepInversion			42.22%	4.04	151.52	0.4051	0.0119
			19.96%	4.29	87.09	0.7483	0.0118
✓			56.69%	5.51	65.12	0.7732	0.1529
✓	✓		58.61%	5.41	**63.36**	**0.7829**	0.1617
✓		✓	59.42%	6.31	65.34	0.6589	0.2845
✓	✓	✓	**60.57%**	**6.44**	63.54	0.6692	**0.2960**

Table 3: Effectiveness of DeepInversion and each component of our method. Our method outperforms DI in IS/FID scores and K.D with a small number of images (50k)

rate. Finally, we evaluate the training accuracy by forwarding the CIFAR-10/100 training set to trained classifiers. Table 2 depicts how the images from NaturalInversion restores the accuracy of model compared to other methods, meaning that our method successfully captures the original training set distribution. Our method outperforms prior works with the training accuracy by a large gap. In addition, we observed that our approach ensures high accuracy, even though the types of training models are different from the inversion model, implying that our method produces more "generalized" images that approximate the original data distribution. The test accuracy result of CIFAR-10/100 is available in the appendix.

Ablation Study

Effectiveness of Each Component in NaturalInversion. We conduct the ablation experiments to understand how each component affects our overall method. We synthesize 50k images from ResNet-34 trained by CIFAR-100 for each configuration. We perform the knowledge distillation to the ResNet-18 student network, and evaluate the quality of synthesized images using FID, IS, and P&R. All settings in image synthesis are the same as Section 4.1. Table 3 reports the result of ablation experiments. The *one-to-one* generator can reduce the mode collapse of synthesized images. Also, ACS parameters per each instance image affect the image's color. These components increase the student network accuracy by improved diversity. FTP leads to synthesizing higher fidelity images by using the feature maps, achieving the best FID and Precision. In summary, FTP improves fidelity, and one-to-one generator and ACS help diversity. The accuracy of the student network achieves the best accuracy, 60.57%, when using all components.

Feature Stage	f_4	f_3,f_4	f_2,f_3,f_4	use all (ours)
1'st stage	0.6892	0.6845	0.6736	0.6634
final stage	77.90	75.84	75.52	73.72

Table 4: FID score under the different usage of feature maps. We gradually choose the feature maps of ResNet-34 from high-level(f_4) to low-level feature maps(f_1).

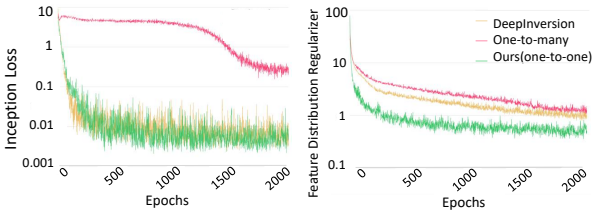

Figure 5: Training loss curve during CIFAR-10 image synthesis. Inception loss (left) and feature distribution regularizer (right). One-to-Many denotes that non-linearity optimization without one-to-one approach.

Bottom-Up Usage of Feature Map. To verify the effectiveness of our *Feature Transfer Pyramid*, we gradually increase feature maps combined to FTP. We observe how close the distribution of synthesized images is to the original distribution according to the number of feature maps. We continuously select high-level and low-level feature maps, then build up the FTP blocks. First, we synthesize 10k images from the ResNet-34 trained CIFAR-10 on each case. Then, we calculate FID scores using the embedding layer of the Inception-v3. As shown in Table 4, we demonstrate that the more we gradually stack FTP block, the lower the FID score. Since each block of FTP enhances the corresponding characteristics of the original target class, we synthesize the images closer to the original data.

Training Loss Curve. To test that our generative model with *one-to-one* approach helps to converge to a specific optimal point faster, we test training loss convergence of three cases: (a) optimization without non-linearity (b) non-linear optimization without *one-to-one* approach (c) our method. Fig.5 shows the training loss curve of $\mathcal{L}_{CE}$ and $\mathcal{R}_{BN}$ that play a key role in generating images in NaturalInversion. We simultaneously plot the training loss curve during 2k CIFAR-10 image inversion epoch. Because of the difficulty of optimizing the input space without non-linearity, DI converges slower than our method, and the final training loss is greater than ours for both $\mathcal{L}_{CE}$ and $\mathcal{R}_{BN}$. Moreover, using the generator to induce non-linearity helps the convergence by greater ability to represent the complex characteristics of the original distribution. In the end, we maximize the improvement in converge speed and training loss when using *one-to-one* generator, leading to eased optimizing process.

Application 1: Knowledge Distillation (KD)

In this section, we ensure that we can transfer information from a teacher network to a student network without origi-

Dataset	Teacher	Student	Accuracy							
			T.	S.	DAFL	DFAD	DI	ADI	ours	A-ours
CIFAR-10	ResNet-34	ResNet-18	95.57	95.20	92.22	93.30	91.43	93.26	93.72	**94.87**
	ResNet-34	VGG-11	95.57	92.44	*44.30	*88.66	*83.49	*75.53	**89.41**	88.46
	ResNet-34	MobileNetV2	95.57	94.69	*72.16	*92.71	*91.00	*93.64	92.96	**94.06**
	VGG-11	ResNet-18	92.44	95.20	*84.19	*89.35	83.82	90.36	90.10	**91.84**
	VGG-11	VGG-11	92.44	92.44	*82.18	***91.34**	84.16	90.78	89.79	91.07
	VGG-11	MobileNetV2	92.44	94.69	*54.79	*84.96	*87.77	*88.97	89.59	**90.62**
CIFAR-100	ResNet-34	ResNet-18	78.02	76.87	**74.47**	67.70	*45.91	*64.38	67.00	72.82
	ResNet-34	VGG-11	78.02	68.64	*48.43	*20.61	*36.04	*51.06	61.96	**65.37**
	ResNet-34	MobileNetV2	78.02	68.02	*59.46	*57.30	*44.30	*58.88	66.42	**71.26**
	VGG-16	ResNet-18	73.75	76.87	*24.91	*53.41	*50.32	*56.36	66.32	**69.84**
	VGG-16	VGG-11	73.75	68.64	*23.96	*44.34	*46.24	*45.14	61.68	**64.37**
	VGG-16	MobileNetV2	73.75	68.02	*9.00	*41.61	*37.76	*54.38	64.04	**66.95**

Table 5: The results of data-free knowledge distillation on CIFAR-10/100 with synthesized images from various inversion methods. The bold type is the best value of the same architecture experiment, and the underbar type is the second-best value. T. and S. means the baseline accuracy of teachers and students trained on the original training set. *: our re-implementations.

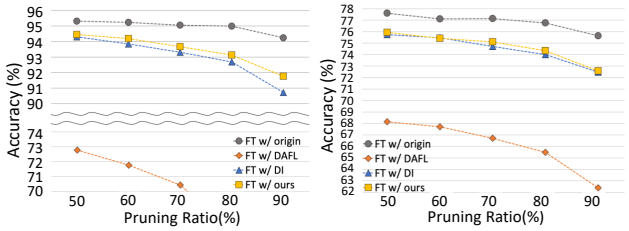

Figure 6: Pruning performance of ResNet-34 on CIFAR-10 (left) and CIFAR-100 (right) under different pruning ratios. We synthesize images from ResNet-34 and fine-tune the pruned model using synthesized images by various methods.

nal data and outperform existing data-free knowledge distillation approaches. We compare our approach with different methods: DAFL, DFAD, DI, and ADI (Yin et al. 2020). Especially, ADI improves upon DI with increment in diversity by the teacher-student disagreement. Since our method can be applied to existing frameworks of inversion, we combine our approach with the ADI method to improve the knowledge distillation performance. For CIFAR-10 KD, we use ResNet-34, VGG-11 as a teacher network, and use ResNet-34 and VGG-16 for CIFAR-100 KD. To verify our method synthesizes less biased images to the teacher, which means generalized images regardless of the model type, we distill knowledge to various students from each teacher. We choose the ResNet-18 (He et al. 2016), VGG-11, and MobileNet-V2 (Sandler et al. 2018) as a student network. We synthesize the 256k images from the teacher network and use all images to train the student network. The results for the data-free knowledge distillation are reported in Tab. 5, showing that our approach achieves comparable performance compared to the most of other methods. Furthermore, although the teacher and student network have different architecture types, NaturalInversion still achieves high performance over other methods, indicating that our method produces less model-biased images regardless of the data and network type.

Application 2 : Pruning

We investigate that our method can improve the pruned model accuracy without real data. For the pruning, we utilized L1-norm pruning criteria that prunes a certain percentage of filters with smaller L1-norm (Liu et al. 2017). We carry out experiments on different pruning ratios from 50% to 90%, and all experiments are performed with ResNet-34. Our pruning setup is the same with Liu et al. (2018), and we locally prune the least important channels in each layer by the same pruning ratio. After pruning ResNet-34 on CIFAR-10/100, we fine-tune the pruned model using synthesized images from the baseline model for 20 epochs by SGD with 0.001 learning rate. As shown in Fig.6, DAFL has poor performance with 58.34% at a pruning ratio 90% on CIFAR-10 because it synthesizes images for a specific purpose, knowledge distillation. DI has low performance on CIFAR-100 because the more complexity the image, the more difficult it is to optimize without non-linear characteristics. In contrast, we achieve the best accuracy recovery of a pruned model up to 11.21% improvement on 90% pruning ratio due to the high fidelity and diversity images. This result ensure that our approach reflects the statistics of original data.

Conclusion

In this paper, we improve the quality of the synthesized image compared to conventional inversion methods with our proposed approach: NaturalInversion. First, we enhanced the characteristics of target class via *Feature Transfer Pyramid* by using multi-scale feature maps from classifier. Second, we used *one-to-one* generator for alleviating the mode collapse problem and bring the non-linearity. Lastly, we proposed *Adaptive Channel Scaling* parameters to implicitly learn the optimal channel scale range which has been learned by the classifier. Through extensive experiments, we demonstrated the effectiveness of NaturalInversion. Our methods not only capture the original data distribution but also are generalized, less biased to inversion model. We hope this work helps for the further progress in synthesizing realistic images in data-free conditions.

Acknowledgements

This work was supported by the National Research Council of Science & Technology (NST) grant by the Korea government (MSIT) [CRC-20-02-KIST], and by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00456, Development of Ultra-high Speech Quality Technology for Remote Multi-speaker Conference System).

References

Chen, H.; Wang, Y.; Xu, C.; Yang, Z.; Liu, C.; Shi, B.; Xu, C.; Xu, C.; and Tian, Q. 2019. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3514–3522.

Chu, X.; Zheng, A.; Zhang, X.; and Sun, J. 2020. Detection in crowded scenes: One proposal, multiple predictions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12214–12223.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Fang, G.; Song, J.; Shen, C.; Wang, X.; Chen, D.; and Song, M. 2019. Data-free adversarial distillation. *arXiv preprint arXiv:1912.11006*.

Gatys, L.; Ecker, A. S.; and Bethge, M. 2015. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28: 262–270.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2414–2423.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Haroush, M.; Hubara, I.; Hoffer, E.; and Soudry, D. 2020. The knowledge within: Methods for data-free model compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8494–8502.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Heitz, E.; Vanhoey, K.; Chambon, T.; and Belcour, L. 2021. A Sliced Wasserstein Loss for Neural Texture Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9412–9420.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. PMLR.

Islam, M. A.; Jia, S.; and Bruce, N. D. 2020. How much position information do convolutional neural networks encode? *arXiv preprint arXiv:2001.08248*.

Kalischek, N.; Wegner, J. D.; and Schindler, K. 2021. In the light of feature distributions: moment matching for Neural Style Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9382–9391.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario.

Lin, T.; Ma, Z.; Li, F.; He, D.; Li, X.; Ding, E.; Wang, N.; Li, J.; and Gao, X. 2021. Drafting and Revision: Laplacian Pyramid Network for Fast High-Quality Artistic Style Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5141–5150.

Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; and Zhang, C. 2017. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, 2736–2744.

Liu, Z.; Sun, M.; Zhou, T.; Huang, G.; and Darrell, T. 2018. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*.

Mahendran, A.; and Vedaldi, A. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5188–5196.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Mordvintsev, A.; Olah, C.; and Tyka, M. 2015. Deepdream-a code example for visualizing neural networks. *Google Research*, 2(5).

Olah, C.; Mordvintsev, A.; and Schubert, L. 2017. Feature visualization. *Distill*, 2(11): e7.

Sajjadi, M. S.; Bachem, O.; Lucic, M.; Bousquet, O.; and Gelly, S. 2018. Assessing generative models via precision and recall. *arXiv preprint arXiv:1806.00035*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29: 2234–2242.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Shocher, A.; Gandelsman, Y.; Mosseri, I.; Yarom, M.; Irani, M.; Freeman, W. T.; and Dekel, T. 2020. Semantic pyramid for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7457–7466.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wang, P.; Li, Y.; Singh, K. K.; Lu, J.; and Vasconcelos, N. 2021. IMAGINE: Image Synthesis by Image-Guided Model Inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3681–3690.

Xu, Y.; Shen, Y.; Zhu, J.; Yang, C.; and Zhou, B. 2021. Generative hierarchical features from synthesizing images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4432–4442.

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8715–8724.

Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; and Lipson, H. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhai, X.; Kolesnikov, A.; Houlsby, N.; and Beyer, L. 2021. Scaling vision transformers. *arXiv preprint arXiv:2106.04560*.