Supplementary Material for ”NAYER: Noisy Layer Data Generation for Efficient and Effective Data-free Knowledge Distillation”

Minh-Tuan Tran[1], Trung Le[1], Xuan-May Le[2], Mehrtash Harandi[1], Quan Hung Tran[3],
Dinh Phung[1]
[1]Monash University, [2]University of Melbourne, [3]ServiceNow
{tuan.tran7,trunglm,mehrtash.harandi,dinh.phung}@monash.edu
xuanmay.le@student.unimelb.edu.au, hungquan.tran@servicenow.com

A. Training Details

A.1. Teacher Model Training Details

In this work, we employed pretrained ResNet-34 and WideResnet-40-2 teacher models from [2] for CIFAR-10 and CIFAR-100. For Tiny ImageNet, we trained ResNet-34 from scratch using PyTorch, and for ImageNet, we utilized the pretrained ResNet-50 from PyTorch. Teacher models were trained with SGD optimizer, initial learning rate of 0.1, momentum of 0.9, and weight decay of 5e-4, using a batch size of 128 for 200 epochs. Learning rate decay followed a cosine annealing schedule.

Table 1. Generator Network ($\mathcal{G}$) Architecture for CIFAR10, CIFAR100 and TinyImageNet.

Output	Size Layers
1000	`Input`
$128 \times h/4 \times w/4$	`Linear, BatchNorm1D, Reshape`
$128 \times h/4 \times w/4$	`SpectralNorm (Conv (3 × 3)), BatchNorm2D, LeakyReLU`
$128 \times h/2 \times w/2$	`UpSample (2×)`
$64 \times h/2 \times w/2$	`SpectralNorm (Conv (3 × 3)), BatchNorm2D, LeakyReLU`
$64 \times h \times w$	`UpSample (2×)`
$3 \times h \times w$	`SpectralNorm (Conv (3 × 3)), Sigmoid, BatchNorm2D`

Table 2. Generator Network ($\mathcal{G}$) Architecture for ImageNet.

Output	Size Layers
1000	`Input`
$128 \times h/16 \times w/16$	`Linear, BatchNorm1D, Reshape`
$128 \times h/16 \times w/16$	`SpectralNorm (Conv (3 × 3)), BatchNorm2D, LeakyReLU`
$128 \times h/8 \times w/8$	`UpSample (2×)`
$128 \times h/8 \times w/8$	`SpectralNorm (Conv (3 × 3)), BatchNorm2D, LeakyReLU`
$128 \times h/4 \times w/4$	`UpSample (2×)`
$64 \times h/4 \times w/4$	`SpectralNorm (Conv (3 × 3)), BatchNorm2D, LeakyReLU`
$64 \times h/2 \times w/2$	`UpSample (2×)`
$64 \times h/2 \times w/2$	`SpectralNorm (Conv (3 × 3)), BatchNorm2D, LeakyReLU`
$64 \times h \times w$	`UpSample (2×)`
$3 \times h \times w$	`SpectralNorm (Conv (3 × 3)), Sigmoid, BatchNorm2D`

A.2. Student Model Training Details

To ensure fair comparisons, we adopt the generator architecture outlined in [2] for all experiments. Specifically, the generator architecture for CIFAR10, CIFAR100, and TinyImageNet is elaborated upon in Table 1, while the generator architecture for ImageNet is provided in Table 2. Across all experiments, we maintain a consistent approach for training the student model, employing a batch size of 512. We utilize the SGD optimizer with a momentum of 0.9 and a variable learning rate, following a cosine annealing schedule that starts at 0.1 and ends at 0, to optimize the student parameters ($\theta_{\mathcal{S}}$). Additionally, we employ the Adam optimizer with a learning rate of 4e-3 for optimizing the generator.We present the results in three distinct variants, each corresponding to a different value of $\mathcal{E}$: 100, 200, and 300, all incorporating a configuration of 20 warm-up epochs, in line with the settings defined in [2]. Further details regarding the parameters can be found in Table 3.

Table 3. The hyperparameters for NAYER applied to four different datasets are detailed below. Specifically, α_{cls}, α_{bn}, and α_{adv} are the hyperparameters associated with Eq. (**??**), and their values are consistent with the settings defined in [2]. The variables I and S denote the number of iterations for generating and training the student, respectively, while g represents the training steps to optimize the generator $\mathcal{G}_{\theta_{\mathcal{G}}}$ and the noisy layers $\mathcal{Z}$.

	batch size (student)	batch size (generator)	α_{cls}	α_{bn}	α_{adv}	I	g	S
CIFAR10	512	400	0.5	10	1.33	2	30	400
CIFAR100	512	400	0.5	10	1.33	2	40	400
TinyImageNet	256	200	0.5	10	1.33	4	60	1000
ImageNet	128	50	0.1	0.1	0.1	20	100	2000

B. Extended Results

B.1. Experiments in Segmentation:

In response to your feedback, we conducted semantic segmentation experiments following FM [1] settings. By utilizing dataset part names such as ’Basements, Bathrooms, ...’ for LTE and the Noisy Layer as the random source, our method in Table 4 outperforms previous works with better IoU.

Method	DFAD	DAFL	FM	NAYER
Synthetic Time	6.0h	3.99h	**0.82h**	**0.82h**
mIoU	0.364	0.105	0.366	**0.385**

Table 4. Mean IoU on NYUv2 Segmentation dataset.

B.2. Noisy Layer Architecture

In Table 5, we compare the different architectures in terms of:

- The averaging accuracy.
- The averaging convergence time, which is the average number of epochs the generator needs to synthesize data with Cross-Entropy (CE) Loss < 0.1.
- The averaging diversity metric, which is calculated using the average L2 distance between the features of new and old data.

The different architectures for the Noisy Layer include:

- A1: `Linear`
- A2: `Linear`, `Linear`
- A3: `BatchNorm`, `Linear`
- A4: `BatchNorm`, `Linear`, `Linear`
- A5: `BatchNorm`, `Linear`, `Sigmoid`
- A6: `BatchNorm`, `Linear`, `Tanh`
- A7: `BatchNorm`, `Linear`, `ReLU`
- A8: `BatchNorm`, `Linear`, `Dropout`

The result demonstrates that:

- The combination of `BatchNorm` and single `Linear` layer produce the best performance.
- The architecture of multi `Linear` layerslayers results in a longer convergence time and subsequently slightly reduces accuracy.
- The `BatchNorm` pplays an important role in improving accuracy, reducing convergence time by increasing the difference between LTEs.
- The activation function such as `ReLU, Sigmoid` and `Tanh` do not improve the performance of our NAYER.

Table 5. The accuracies of our NAYER and FM (which uses random noise as the input) with varying training steps for generators.

	A1	A2	A3	A4	A5	A6	A7	A8
Avg. Convergence Time	16.58	22.72	9.53	15.73	9.63	9.59	9.61	12.72
Diversity Score	0.131	0.137	0.139	0.141	0.137	0.135	0.138	0.138
Accuracy	92.25	92.11	**93.48**	93.37	93.41	93.37	93.42	93.42

B.3. Comparison with Different Generation Steps

We compare NAYER and FM, both utilizing random noise as input, while adjusting the training steps for their generators. It's important to note that for a fair comparison, we employ the same generator architectures, including the additional linear layer (noisy layer for NAYER) for FM. Furthermore, all models are trained for 300 epochs. This approach allows us to assess their performance under consistent conditions and understand how varying the generator training steps impact their accuracy. The results indicate that our method has the best results with 40 generation steps for CIFAR100 and 30 steps for CIFAR10. Furthermore, NAYER outperforms FM in all cases of generator training steps.

Table 6. The accuracies of our NAYER and FM (which uses random noise as the input) with varying training steps for generators.

Generator's training steps	$g = 2$	$g = 5$	$g = 10$	$g = 20$	$g = 30$	$g = 40$	$g = 50$
FM	57.08	63.83	65.12	66.82	67.51	68.23	68.18
NAYER	**59.23**	**65.14**	**68.13**	**69.31**	**70.42**	**71.72**	**71.70**

B.4. Robust experiments

Thanks for your comments. The robust experiments in three runs in Table 7 shows our method's consistently better accuracy with only minor standard deviation. Notably, previous works omitted these numbers, and due to their high complexity, we did not replicate their results in this rebuttal period.

Table 7. Averaging accuracy and standard deviation in three runs.

	CIFAR10			CIFAR100		
	R34/R18	W402/W162	W402/W161	R34/R18	W402/W162	W402/W161
SpaceshipNet	**95.39**	93.25	90.38	77.41	69.95	58.06
NAYER ($\mathcal{E} = 300$)	95.24 ± 0.15	**94.11** ± **0.18**	**91.94** ± **0.15**	**77.56** ± **0.12**	**71.72** ± **0.14**	**62.23** ± **0.21**

B.5. NAYER without Label Text Embedding (LTE)

To highlight our method's LTE-independent capability, we conducted experiments using one-hot vectors and Noisy Layer in Table 8. Despite the lower accuracy of the one-hot version compared to the LTE version, our method still outperforms the SOTA approach in both scenarios.

Table 8. Accuracy in CIFAR10 with W402/W162 Architecture.

Method	SpaceshipNet	NAYER with LTE	NAYER with one-hot vector
CIFAR10	93.25	94.07	93.72
CIFAR100	69.95	71.72	70.78

B.6. Non-BatchNorm Architecture

The need for batch norm loss is the limitation for most SOTA DFKD methods. Therefore, exploring high performance with non-batchnorm architectures is an intriguing future direction. For this rebuttal, we conduct the CIFAR10 experiments with AlexNet as the student (Table 9). The results suggest that our NAYER outperforms previous work when applied to AlexNet.

Table 9. Accuracy with AlexNet Student.

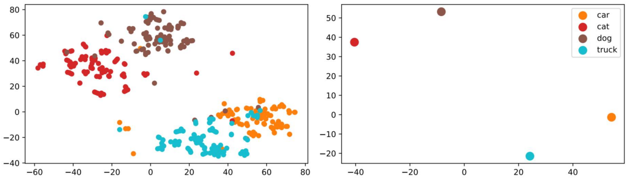
	Teacher Accuracy	Student Accuracy	FM ($\mathcal{E} = 300$)	NAYER ($\mathcal{E} = 300$)
AlexNet/AlexNet	74.74%	74.74%	65.37%	70.14%
Resnet34/AlexNet	95.70%	74.74%	68.38%	71.15%

B.7. Additional Abalation Studies for Noisy Layer

Inspired by your recommendations, we conducted additional experiments in Table 10. The results show that: (4.1) NL with reinitialization (wRI) outperforms with out reinitialization (woRI); (4.2) With beta greater than one, the sum method performs worse than the NL; (4.3) While we used normal noise for the sum method, we further experimented with uniform noise (uni); however, the results remained significantly lower than our NL; (4.4) We will include the note of a zero bias in our revised paper.

Table 10. Additional ablation Study for Noisy Layer

Method	NL(woRI)	NL(wRI)	sum(1.5)	sum(2.0)	sum(3.0)	uni(1.0)	uni(1.5)	uni(2.0)
Avg. Convergence Time(↓)	8.68	9.53	-	-	-	-	-	-
Diversity Score(↑)	0.016	0.139	0.129	0.133	0.135	0.113	0.128	0.139
Accuracy(↑)	14.82	**93.48**	90.23	90.15	89.12	87.75	88.21	87.15



(a) t-SNE of CIFAR10 data

(b) t-SNE of LTE

Figure 1. t-SNE Visualization of Label-Text Embedding and Ground-Truth Dataset Distribution for Four Classes: Car, Cat, Dog, and Truck.

B.8. t-SNE Visuallization of LTE and Ground-truth Dataset Distribution

In this section, we aim to illustrate the interclass information captured by LTE (Label-Text Embedding). To achieve this, we provide t-SNE visualizations of the embeddings for labels and ground-truth data distribution pertaining to four distinct classes: Car, Cat, Dog, and Truck. The t-SNE representation of LTE closely aligns with the ground-truth distribution, especially in the proximity between classes like Car and Truck, as well as Cat and Dog, indicating notably smaller distances compared to other class pairings.

B.9. Visualization.

The synthetic results achieved by NAYER within just 100 generator training steps on ImageNet by employing the ResNet-50 as teacher model are presented in Figure 2a-b. For further comparison, we also visualize synthetic images generated by NAYER, FM, CMI, and DeepInv in Figure 2c-f. All of these samples are generated using 20 steps with a ResNet-34 teacher model in the CIFAR-10 dataset. While it remains challenging for human recognition and significantly differs from real datasets, our synthetic images contain common knowledge that represents the classes, thereby visibly demonstrating superior quality compared to other methods.

C. Further Discussion

Does NAYER Preserve a Data-free Setting? In the definition, data-free knowledge distillation is characterized by training a model without direct access to the teacher model's training data, stemming from the necessity for privacy in datasets. Therefore, the introduction of static label text embeddings does not violate this definition as it does not interact with any training data. In real-world applications, obtaining label embeddings from publicly available pretrained language models like CLIP or ChatGPT is quick and straightforward. Crucially, our method requires no fine-tuning or retraining of these pretrained models, and thus it does not use any external data. Consequently, it can seamlessly adapt to any real-world DFKD application.

Does NAYER Work in Meaningless Label Datasets? The paper suggests the utilization of label-text embedding (LTE), acknowledging its potential drawbacks in datasets lacking meaningful labels, such as those involving chemical compounds. However, our approach highlights two key advantages associated with LTE. Firstly, LTE acts as a dense vector, encompassing more information and thereby facilitating a smoother learning process for the model. Secondly, LTE has the ability to depict relationships between classes. In cases where a class lacks a meaningful label, we can still utilize the class index to generate a label text, such as `"a class of {class_index}"`, enabling the extraction of LTE.

While this approach may not completely capture the relationships between classes, it provides richer information compared to a one-hot vector. To support this assertion, we conducted an ablation study on various prompt engineering techniques in Section 4.4. The results indicate that when using only the label index instead of the label name, the performance of P3 remains significantly superior to the best baseline. Specifically, this template P3 achieves 93.72% accuracy compared to SpaceshipNet (the state-of-the-art model) with 93.25%, and 71.17% compared to 69.95%, respectively. This underscores the viability of employing the label index, particularly in datasets with less meaningful labels, further validating the effectiveness of our methods in real-world applications.

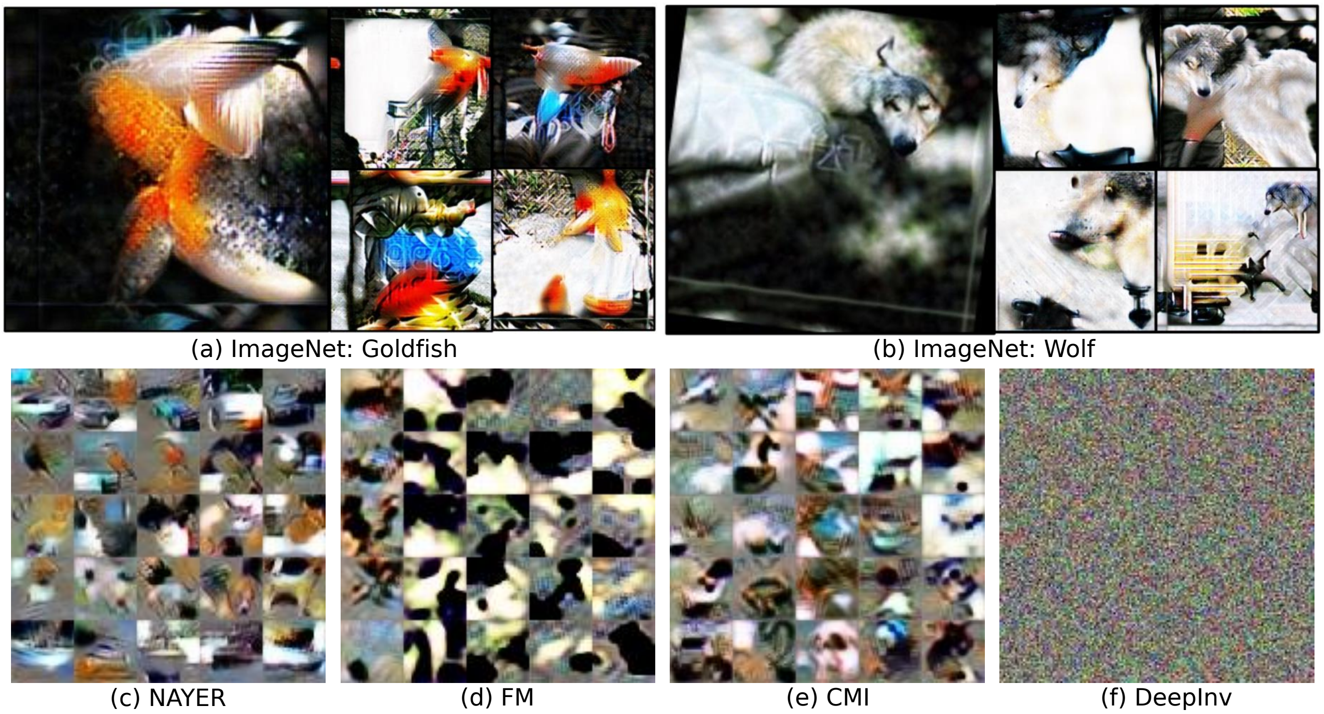
D. Furture Works

The proposed NAYER does not incorporate the innovative techniques utilized in current SOTA methods, such as feature mixup [4], knowledge acquisition and retention [3], and momentum updating [1]. This leaves space for potential improvements through the integration of these techniques in the future. Additionally, NAYER can be applied to various data-free methods, including but not limited to data-free quantization or data-free model stealing.

References

[1] Kien Do, Thai Hung Le, Dung Nguyen, Dang Nguyen, Haripriya Harikumar, Truyen Tran, Santu Rana, and Svetha Venkatesh. Momentum adversarial distillation: Handling large distribution shifts in data-free knowledge distillation. *Advances in Neural Information Processing Systems*, 35:10055–10067, 2022. 3

[2] Gongfan Fang, Kanya Mo, Xinchao Wang, Jie Song, Shitao Bei, Haofei Zhang, and Mingli Song. Up to 100x faster data-free knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6597–6604, 2022. 1


(a) ImageNet: Goldfish

(b) ImageNet: Wolf

(c) NAYER

(d) FM

(e) CMI

(f) DeepInv

Figure 2. (a, b) Display synthetic data generated by our NAYER for ImageNet in just 100 steps. (c, d, e, f) Showcase synthetic data generated for 5 classes (from top to bottom: Car, Bird, Cat, Dog, Ship) in CIFAR10, using only 20 steps of NAYER, FM, CMI, and DeepInv.

[3] Gaurav Patel, Konda Reddy Mopuri, and Qiang Qiu. Learning to retain while acquiring: Combating distribution-shift in adversarial data-free knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7786–7794, 2023. 3

[4] Shikang Yu, Jiachen Chen, Hu Han, and Shuqiang Jiang. Data-free knowledge distillation via feature exchange and activation region constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24266–24275, 2023. 3