2022 VDS Lab Seminar

Dataset Distillation

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
Junho Park

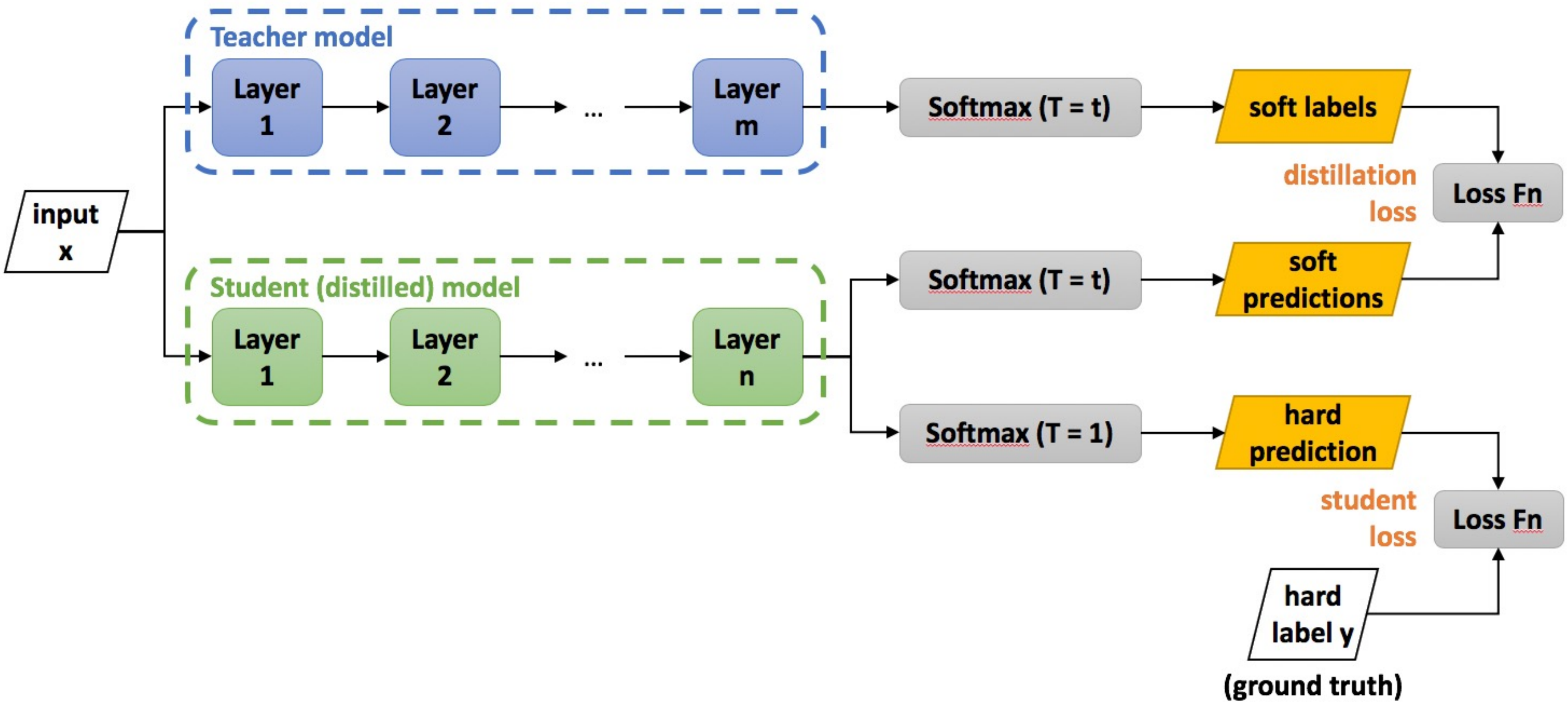
Outline

- Background
  - Knowledge Distillation
    - Hinton et al. In 2015 NIPS Workshop.
  - Dataset Distillation
    - Wang et al. In 2018 arXiv.
- Paper
  - Dataset Distillation by Matching Training Trajectories
    - Cazenavette et al. In 2022 CVPR (oral).
- Conclusion
  - Discussion
  - Limitations

1) Geoffrey Hinton, Oriol Vinyals, Jeff Dean. Distilling the Knowledge in a Neural Network. In NIPS 2015 Deep Learning Workshop.

Knowledge Distillation[1)]

- Model compression method in which a small model is trained to mimic a pre-trained, larger model
  - Referred to Teacher-student model
    - Teacher : Large model
    - Student : Small model



서강대학교 SOGANG UNIVERSITY

VDS LAB

1) Geoffrey Hinton, Oriol Vinyals, Jeff Dean. Distilling the Knowledge in a Neural Network. In NIPS 2015 Deep Learning Workshop.

Knowledge Distillation[1)]

- Model compression method in which a small model is trained to mimic a pre-trained, larger model
  - Softmax Temperature

$$p_i = \frac{exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}$$

dog

	cow	dog	cat	car
original hard targets	0	1	0	0
output of geometric ensemble	10^{-6}	.9	.1	10^{-9}
softened output of ensemble	.05	.3	.2	.005

1) Geoffrey Hinton, Oriol Vinyals, Jeff Dean. Distilling the Knowledge in a Neural Network. In NIPS 2015 Deep Learning Workshop.

Knowledge Distillation[1)]

- Model compression method in which a small model is trained to mimic a pre-trained, larger model
  - Softmax Temperature

$$p_i = \frac{exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}$$

dog

cow	dog	cat	car	
0	1	0	0	original hard targets

cow	dog	cat	car	
10^{-6}	.9	.1	10^{-9}	output of geometric ensemble

cow	dog	cat	car	
.05	.3	.2	.005	softened output of ensemble

Knowledge Distillation[1)]

- Model compression method in which a small model is trained to mimic a pre-trained, larger model
  - Softmax Temperature

$$p_i = \frac{exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}$$

dog

cow	dog	cat	car	
0	1	0	0	original hard targets

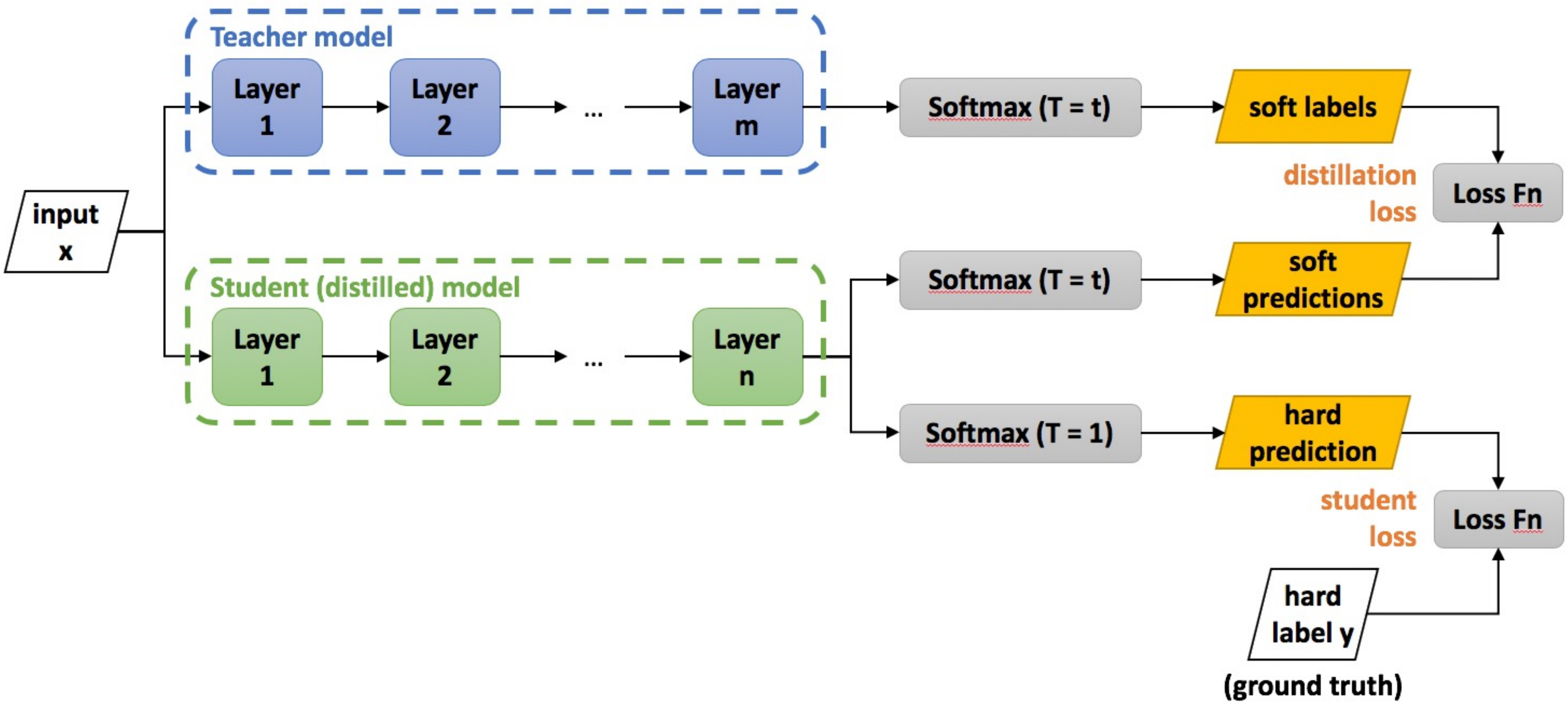
cow	dog	cat	car	
10^{-6}	.9	.1	10^{-9}	output of geometric ensemble

cow	dog	cat	car	
.05	.3	.2	.005	softened output of ensemble

Knowledge Distillation[1)]

- Model compression method in which a small model is trained to mimic a pre-trained, larger model

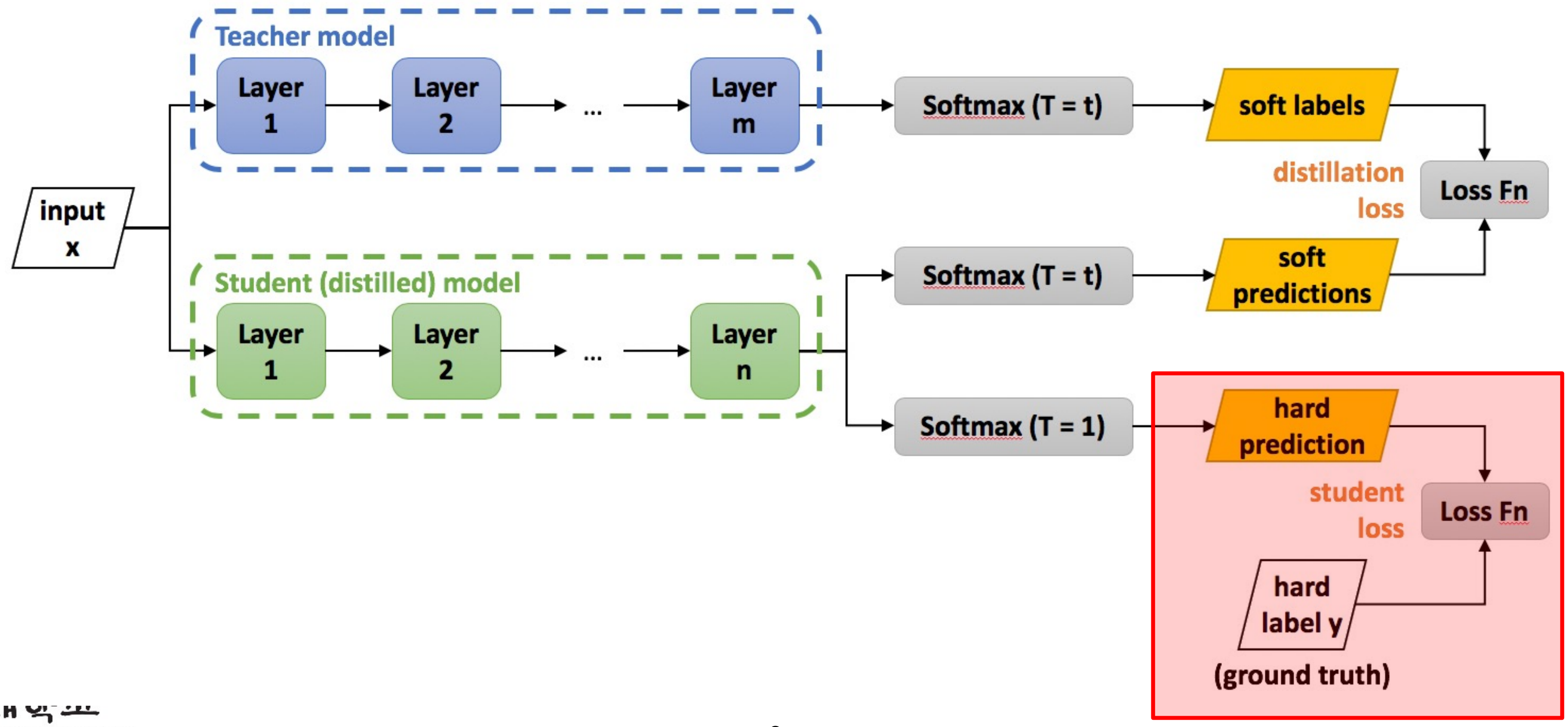
$$\mathcal{L}(x; W) = \alpha * \mathcal{H}(y, \sigma(z_s; T = 1)) + \beta * \mathcal{H}(\sigma(z_t; T = \tau), \sigma(z_s, T = \tau))$$



Knowledge Distillation[1)]

- Model compression method in which a small model is trained to mimic a pre-trained, larger model

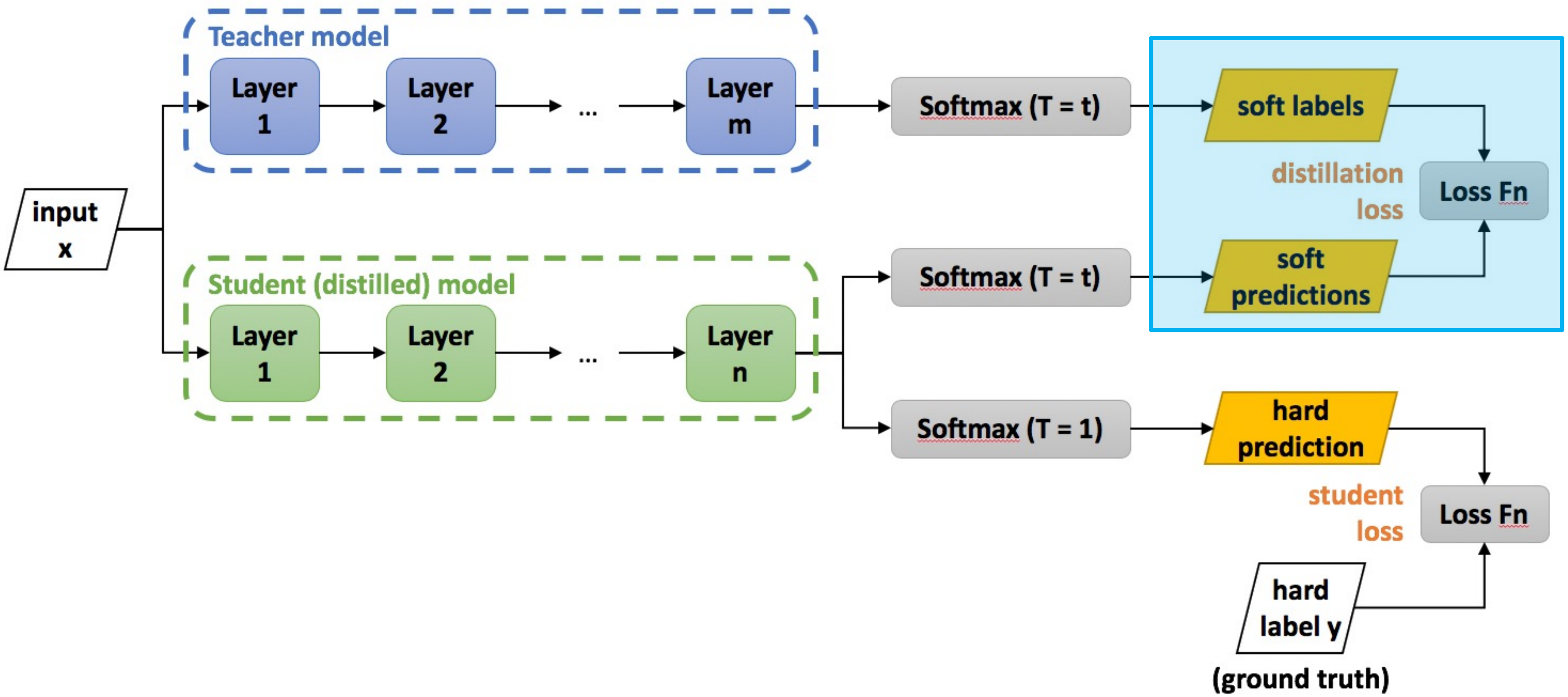
$$\mathcal{L}(x; W) = \alpha * \boxed{\mathcal{H}(y, \sigma(z_s; T=1))} + \beta * \mathcal{H}(\sigma(z_t; T=\tau), \sigma(z_s, T=\tau))$$



Knowledge Distillation[1)]

- Model compression method in which a small model is trained to mimic a pre-trained, larger model

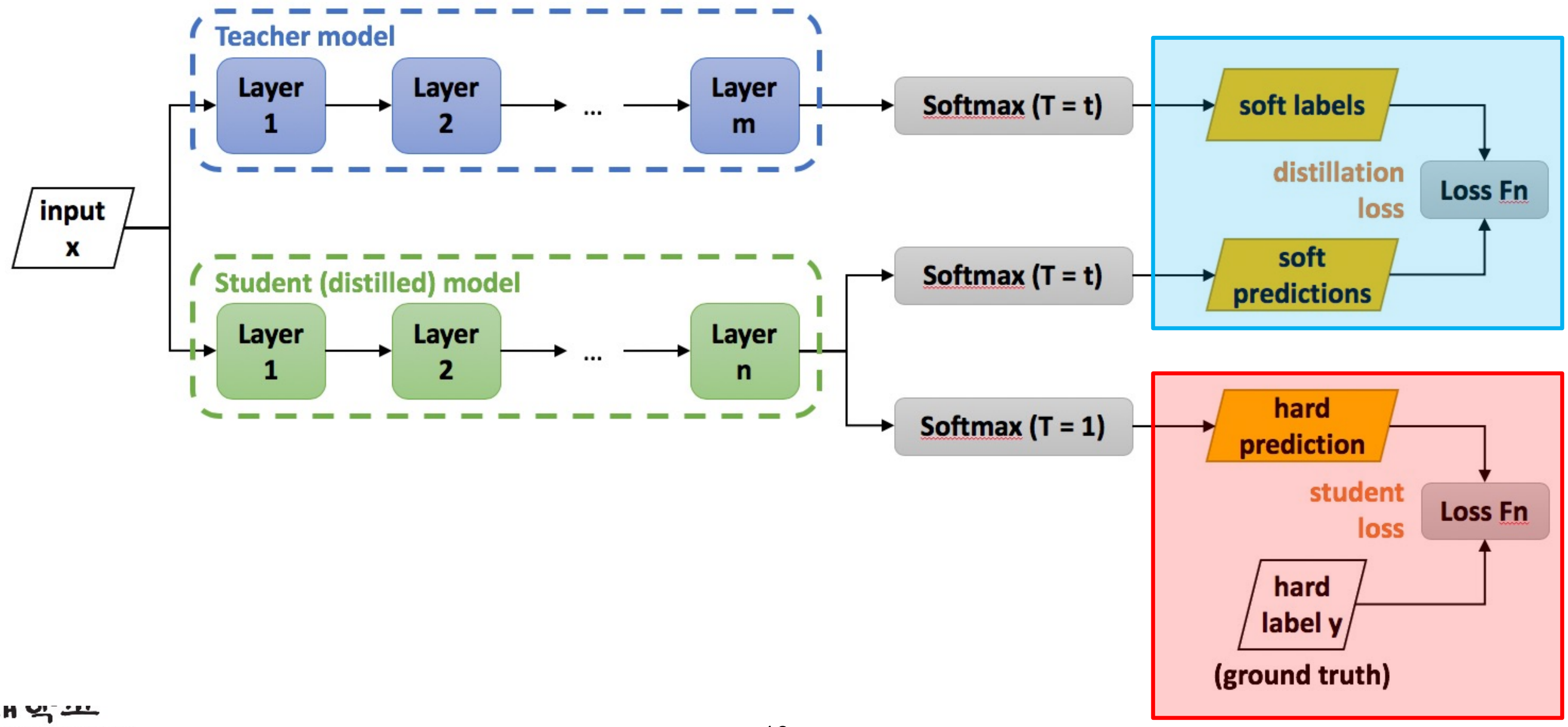
$$\mathcal{L}(x;W) = \alpha * \mathcal{H}(y, \sigma(z_s; T=1)) + \beta * \boxed{\mathcal{H}(\sigma(z_t; T=\tau), \sigma(z_s, T=\tau))}$$



Knowledge Distillation[1)]

- Model compression method in which a small model is trained to mimic a pre-trained, larger model

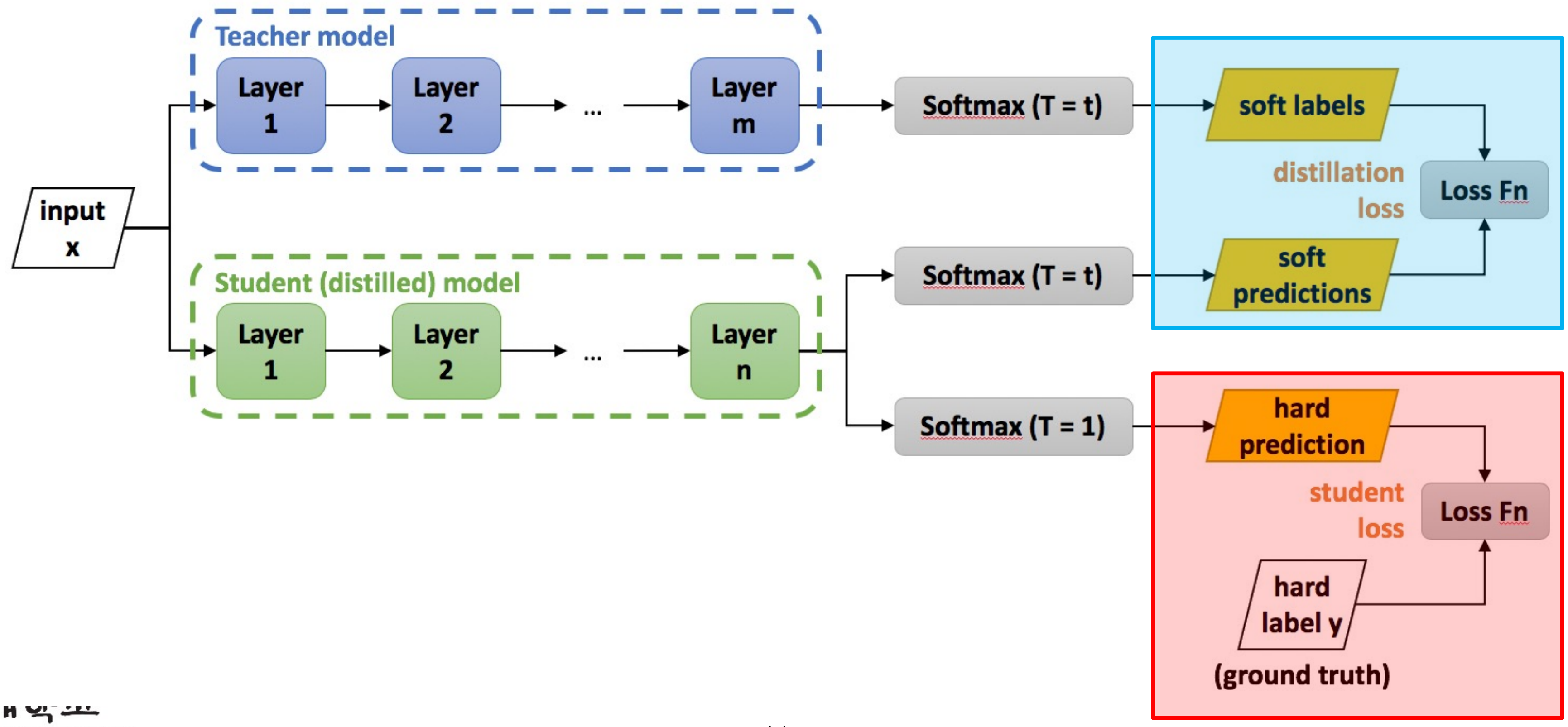
$$\mathcal{L}(x; W) = \alpha * \boxed{\mathcal{H}(y, \sigma(z_s; T=1))} + \beta * \boxed{\mathcal{H}(\sigma(z_t; T=\tau), \sigma(z_s, T=\tau))}$$



Knowledge Distillation[1)]

- Model compression method in which a small model is trained to mimic a pre-trained, larger model

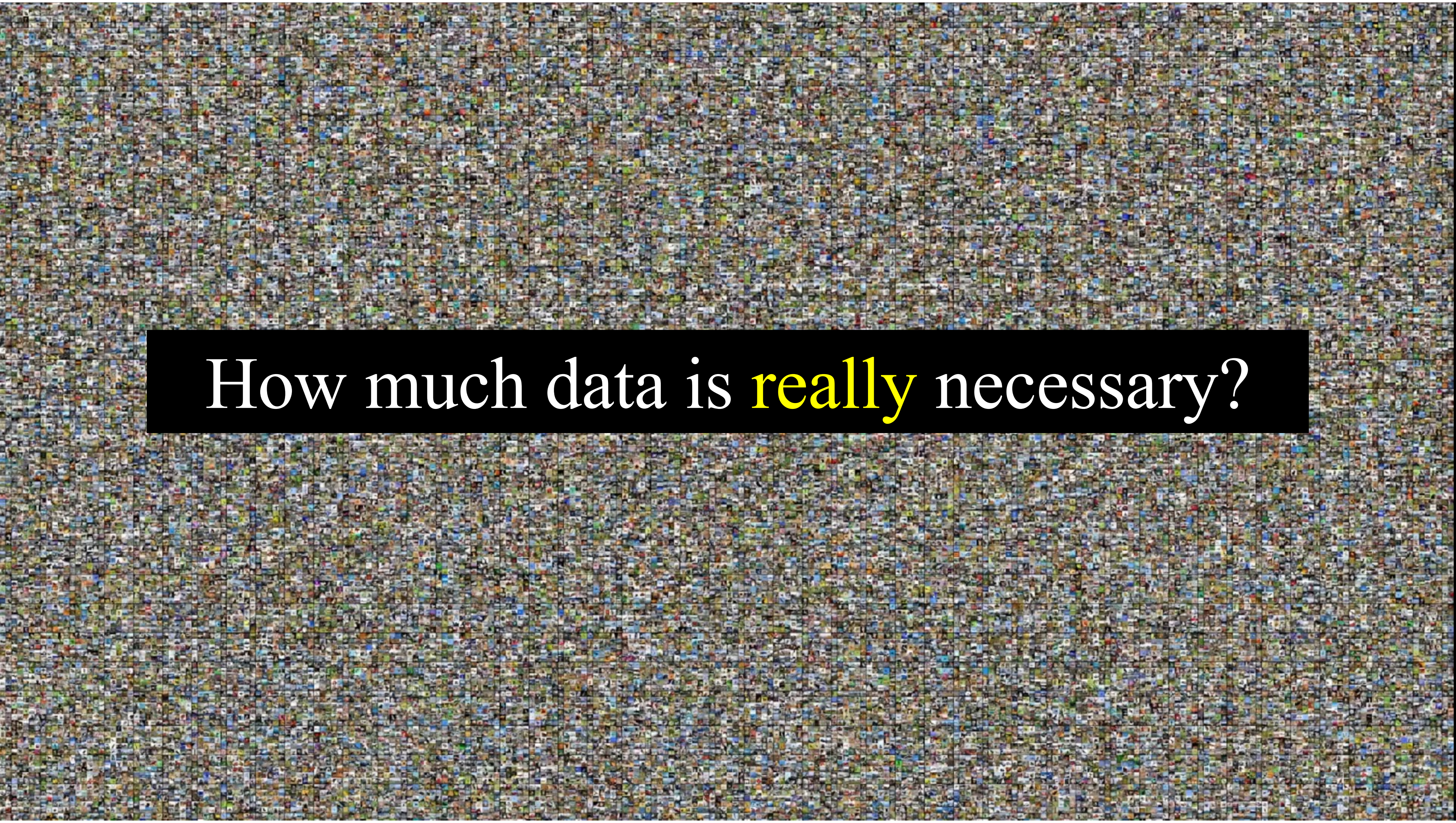
$$\mathcal{L}(x; W) = \alpha * \boxed{\mathcal{H}(y, \sigma(z_s; T = 1))} + \beta * \boxed{\mathcal{H}(\sigma(z_t; T = \tau), \sigma(z_s, T = \tau))}$$



Dataset Distillation[2)]

2) Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, Alexei A. Efros. Dataset Distillaion. In arXiv preprint 2018.

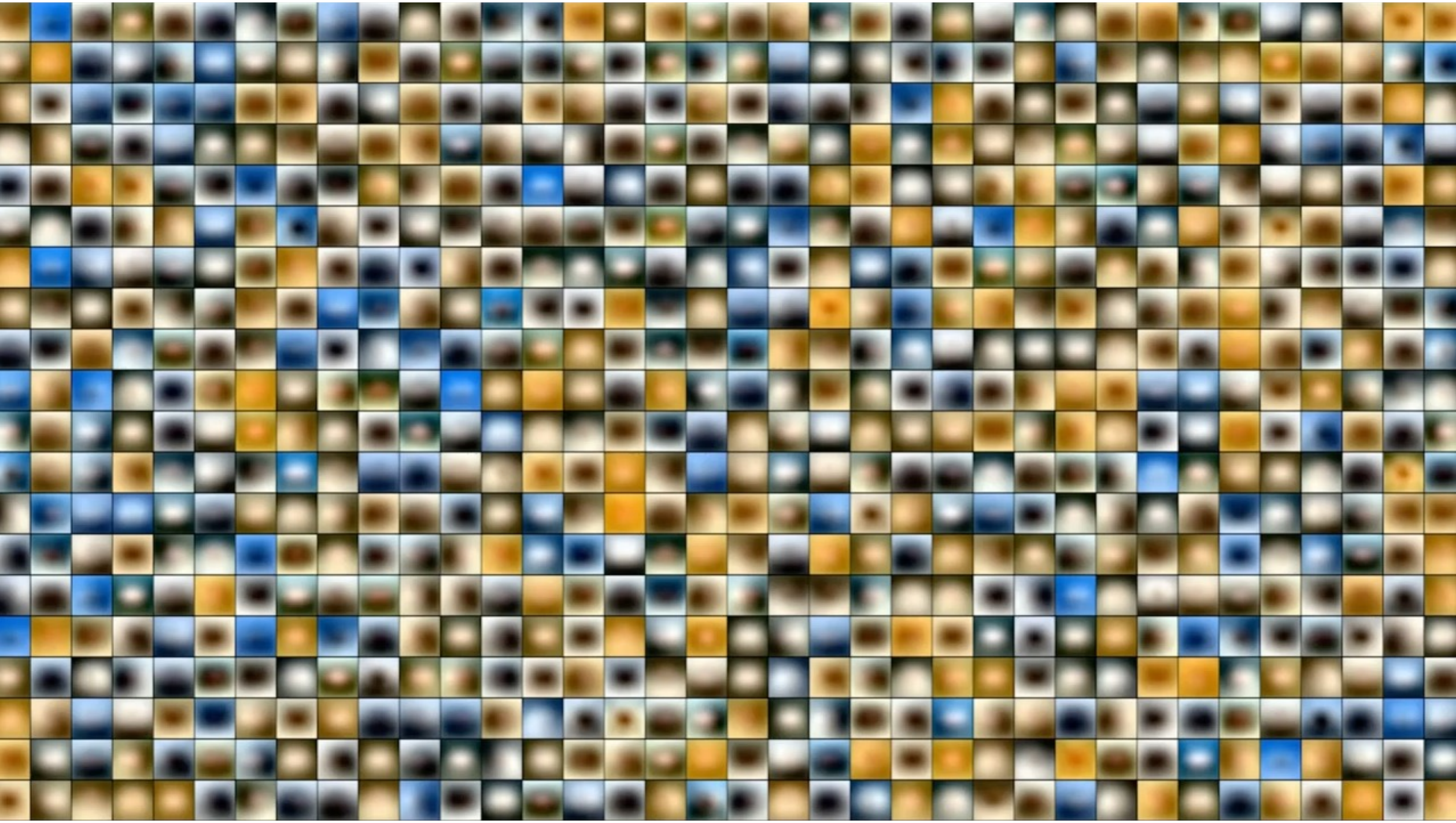
Dataset Distillation[2)]



2) Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, Alexei A. Efros. Dataset Distillaion. In arXiv preprint 2018.

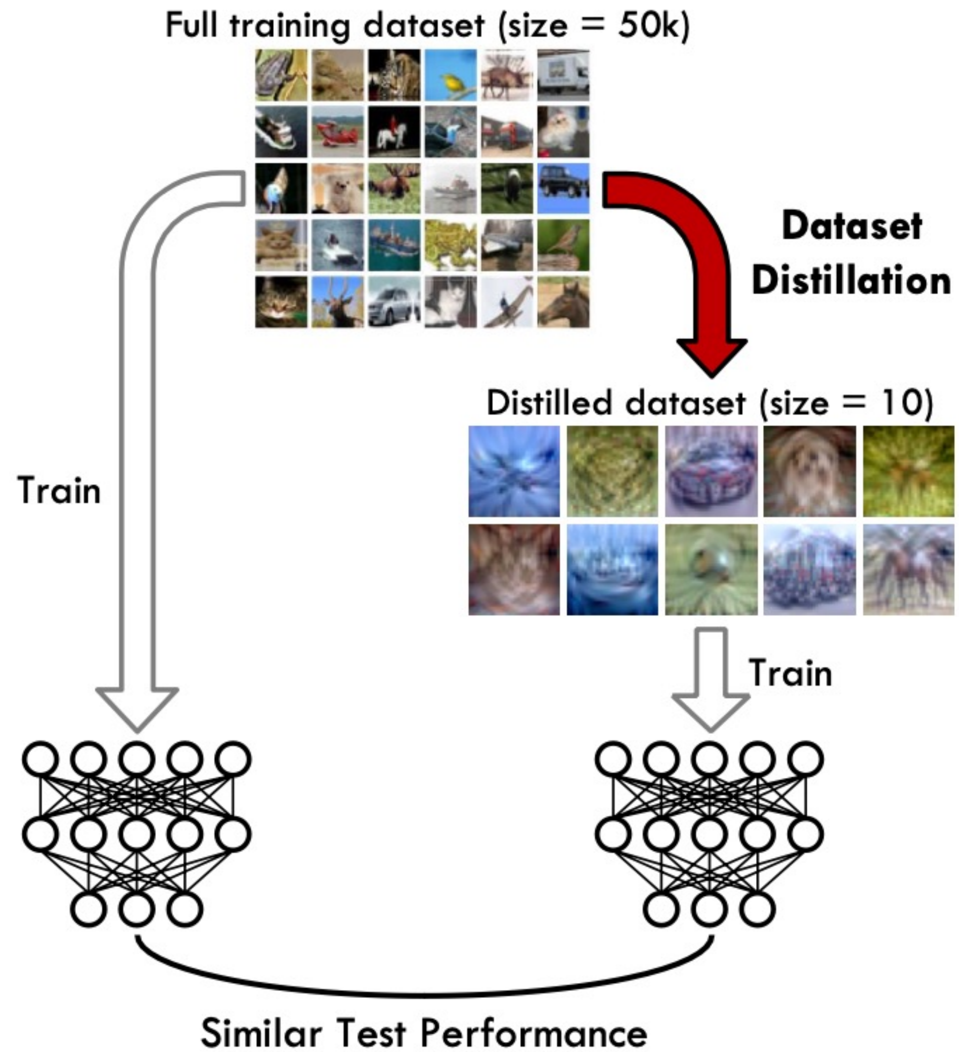
Dataset Distillation[2)]

Dataset Distillation[2)]



Dataset Distillation[2)]

- Idea
  - Not distilling the model,
  - But distilling the dataset.
- Goal
  - Distill the knowledge from a large training dataset into a very small set of synthetic training images.
  - Training a model on the distilled data would give a similar test performance as training one on the original dataset.



2) Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, Alexei A. Efros. Dataset Distillaion. In arXiv preprint 2018.

Dataset Distillation[2)]

- Algorithm
  - To obtain a new, much-reduced synthetic dataset which performs almost as well as the original dataset.

Algorithm 1 Dataset Distillation

Input: $p(\theta_0)$: distribution of initial weights; M: the number of distilled data
Input: α: step size; n: batch size; T: the number of optimization iterations; $\tilde{\eta}_0$: initial value for $\tilde{\eta}$

```
Initialize x̃ = {x̃_i}_{i=1}^M randomly, η̃ ← η̃_0
for each training step t = 1 to T do
    Get a minibatch of real training data x_t = {x_{t,j}}_{j=1}^n
    Sample a batch of initial weights θ_0^(j) ~ p(θ_0)
    for each sampled θ_0^(j) do
        Compute updated parameter with GD: θ_1^(j) = θ_0^(j) − η̃ ∇_{θ_0^(j)} ℓ(x̃, θ_0^(j))
        Evaluate the objective function on real training data: L^(j) = ℓ(x_t, θ_1^(j))
    end for
    Update x̃ ← x̃ − α ∇_x̃ Σ_j L^(j), and η̃ ← η̃ − α ∇_η̃ Σ_j L^(j)
end for
```

- Initialize $\tilde{\mathbf{x}} = \{\tilde{x}_i\}_{i=1}^{M}$ randomly, $\tilde{\eta} \leftarrow \tilde{\eta}_0$
- **for each** training step $t = 1$ to T **do**
  - Get a minibatch of real training data $\mathbf{x}_t = \{x_{t,j}\}_{j=1}^{n}$
  - Sample a batch of initial weights $\theta_0^{(j)} \sim p(\theta_0)$
  - **for each** sampled $\theta_0^{(j)}$ **do**
    - Compute updated parameter with GD: $\theta_1^{(j)} = \theta_0^{(j)} - \tilde{\eta}\nabla_{\theta_0^{(j)}}\ell(\tilde{\mathbf{x}}, \theta_0^{(j)})$
    - Evaluate the objective function on real training data: $\mathcal{L}^{(j)} = \ell(\mathbf{x}_t, \theta_1^{(j)})$
  - **end for**
  - Update $\tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} - \alpha\nabla_{\tilde{\mathbf{x}}}\sum_j \mathcal{L}^{(j)}$, and $\tilde{\eta} \leftarrow \tilde{\eta} - \alpha\nabla_{\tilde{\eta}}\sum_j \mathcal{L}^{(j)}$
- **end for**

Output: distilled data $\tilde{\mathbf{x}}$ and optimized learning rate $\tilde{\eta}$

2) Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, Alexei A. Efros. Dataset Distillaion. In arXiv preprint 2018.

Dataset Distillation[2)]

- Algorithm
  - To obtain a new, much-reduced synthetic dataset which performs almost as well as the original dataset.

Algorithm 1 Dataset Distillation

Input: $p(\theta_0)$: distribution of initial weights; M: the number of distilled data

Input: α: step size; n: batch size; T: the number of optimization iterations; $\tilde{\eta}_0$: initial value for $\tilde{\eta}$

Initialize $\tilde{\mathbf{x}} = \{\tilde{x}_i\}_{i=1}^{M}$ randomly, $\tilde{\eta} \leftarrow \tilde{\eta}_0$

for each training step $t = 1$ to T **do**

  Get a minibatch of real training data $\mathbf{x}_t = \{x_{t,j}\}_{j=1}^{n}$

  Sample a batch of initial weights $\theta_0^{(j)} \sim p(\theta_0)$

  for each sampled $\theta_0^{(j)}$ **do**

    Compute updated parameter with GD: $\theta_1^{(j)} = \theta_0^{(j)} - \tilde{\eta}\nabla_{\theta_0^{(j)}} \ell(\tilde{\mathbf{x}}, \theta_0^{(j)})$

    Evaluate the objective function on real training data: $\mathcal{L}^{(j)} = \ell(\mathbf{x}_t, \theta_1^{(j)})$

  end for

  Update $\tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} - \alpha\nabla_{\tilde{\mathbf{x}}} \sum_j \mathcal{L}^{(j)}$, and $\tilde{\eta} \leftarrow \tilde{\eta} - \alpha\nabla_{\tilde{\eta}} \sum_j \mathcal{L}^{(j)}$

end for

Output: distilled data $\tilde{\mathbf{x}}$ and optimized learning rate $\tilde{\eta}$

Dataset Distillation[2)]

- Algorithm
  - To obtain a new, much-reduced synthetic dataset which performs almost as well as the original dataset.

Algorithm 1 Dataset Distillation

Input: $p(\theta_0)$: distribution of initial weights; M: the number of distilled data
Input: α: step size; n: batch size; T: the number of optimization iterations; $\tilde{\eta}_0$: initial value for $\tilde{\eta}$

```
Initialize x̃ = {x̃_i}_{i=1}^M randomly, η̃ ← η̃_0
for each training step t = 1 to T do
    Get a minibatch of real training data x_t = {x_{t,j}}_{j=1}^n
    Sample a batch of initial weights θ_0^(j) ~ p(θ_0)
    for each sampled θ_0^(j) do
        Compute updated parameter with GD: θ_1^(j) = θ_0^(j) − η̃ ∇_{θ_0^(j)} ℓ(x̃, θ_0^(j))
        Evaluate the objective function on real training data: L^(j) = ℓ(x_t, θ_1^(j))
    end for
    Update x̃ ← x̃ − α∇_x̃ Σ_j L^(j), and η̃ ← η̃ − α∇_η̃ Σ_j L^(j)
end for
```

Output: distilled data $\tilde{\mathbf{x}}$ and optimized learning rate $\tilde{\eta}$

2) Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, Alexei A. Efros. Dataset Distillaion. In arXiv preprint 2018.

Dataset Distillation[2)]

- Algorithm
  - To obtain a new, much-reduced synthetic dataset which performs almost as well as the original dataset.

Algorithm 1 Dataset Distillation

Input: $p(\theta_0)$: distribution of initial weights; M: the number of distilled data
Input: α: step size; n: batch size; T: the number of optimization iterations; $\tilde{\eta}_0$: initial value for $\tilde{\eta}$

Initialize $\tilde{\mathbf{x}} = \{\tilde{x}_i\}_{i=1}^{M}$ randomly, $\tilde{\eta} \leftarrow \tilde{\eta}_0$
for each training step $t = 1$ to T **do**
    Get a minibatch of real training data $\mathbf{x}_t = \{x_{t,j}\}_{j=1}^{n}$
    Sample a batch of initial weights $\theta_0^{(j)} \sim p(\theta_0)$
    for each sampled $\theta_0^{(j)}$ **do**
        Compute updated parameter with GD: $\theta_1^{(j)} = \theta_0^{(j)} - \tilde{\eta}\nabla_{\theta_0^{(j)}} \ell(\tilde{\mathbf{x}}, \theta_0^{(j)})$
        Evaluate the objective function on real training data: $\mathcal{L}^{(j)} = \ell(\mathbf{x}_t, \theta_1^{(j)})$
    end for
    $\boxed{\text{Update } \tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} - \alpha\nabla_{\tilde{\mathbf{x}}} \sum_j \mathcal{L}^{(j)}, \text{ and } \tilde{\eta} \leftarrow \tilde{\eta} - \alpha\nabla_{\tilde{\eta}} \sum_j \mathcal{L}^{(j)}}$
end for
Output: distilled data $\tilde{\mathbf{x}}$ and optimized learning rate $\tilde{\eta}$

Dataset Distillation[2)]

- Algorithm
  - To obtain a new, much-reduced synthetic dataset which performs almost as well as the original dataset.

Algorithm 1 Dataset Distillation

Input: $p(\theta_0)$: distribution of initial weights; M: the number of distilled data
Input: α: step size; n: batch size; T: the number of optimization iterations; $\tilde{\eta}_0$: initial value for $\tilde{\eta}$

```
Initialize x̃ = {x̃_i}_{i=1}^M randomly, η̃ ← η̃_0
for each training step t = 1 to T do
    Get a minibatch of real training data x_t = {x_{t,j}}_{j=1}^n
    Sample a batch of initial weights θ_0^(j) ~ p(θ_0)
    for each sampled θ_0^(j) do
        Compute updated parameter with GD: θ_1^(j) = θ_0^(j) − η̃ ∇_{θ_0^(j)} ℓ(x̃, θ_0^(j))
        Evaluate the objective function on real training data: L^(j) = ℓ(x_t, θ_1^(j))
    end for
    Update x̃ ← x̃ − α∇_x̃ Σ_j L^(j), and η̃ ← η̃ − α∇_η̃ Σ_j L^(j)
end for
```

Output: distilled data $\tilde{\mathbf{x}}$ and optimized learning rate $\tilde{\eta}$

Dataset Distillation[2)]

- History

Dataset Distillation, 2018, arXiv

2) Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, Alexei A. Efros. Dataset Distillaion. In arXiv preprint 2018.

Dataset Distillation[2)]

- History

Dataset Distillation, 2018, arXiv

2020

Flexible Dataset Distillation: Learn Labels instead of Images (NIPS Workshop)

Dataset Condensation with Gradient Matching (ICLR)

Dataset Distillation[2)]

- History

Dataset Distillation, 2018, arXiv

2020

Flexible Dataset Distillation: Learn Labels instead of Images (NIPS Workshop)

Dataset Condensation with Gradient Matching (ICLR)

2021

Dataset Condensation with Differentiable Siamese Augmentation (ICML)

Dataset Distillation with Infinitely Wide Convolutional Networks (NIPS)

Dataset Distillation[2)]

- History

Dataset Distillation, 2018, arXiv

2020

Flexible Dataset Distillation: Learn Labels instead of Images (NIPS Workshop)

Dataset Condensation with Gradient Matching (ICLR)

2021

Dataset Condensation with Differentiable Siamese Augmentation (ICML)

Dataset Distillation with Infinitely Wide Convolutional Networks (NIPS)

2022

Dataset Condensation with Distribution Matching (ICLR)

Dataset Condensation via Efficient Synthetic-Data Parameterization (ICML)

Dataset Distillation[2)]

- History

Dataset Distillation, 2018, arXiv

2020

Flexible Dataset Distillation: Learn Labels instead of Images (NIPS Work… op)

Dataset Condensation with Gradient Matching (ICLR)

2021

Dataset Condens… …ferentiable Siamese… …on (ICML)

Dataset Distillati… …Wide Conv… …s (NIPS)

2022

Datas… …ondensation with Distribution Matching (ICLR)

Da… …Condensation via Efficient Synthetic-Data Parameterization (ICML)

Toy Dataset?

Theoretical Interest?

Dataset Distillation[2)]

- History

Dataset Distillation, 2018, arXiv

2020
- Flexible Dataset Distillation: Learn Labels instead of Images (NIPS Workshop)
- Dataset Condensation with Gradient Matching (ICLR)

2021
- Dataset Condensation with Differentiable Siamese Augmentation (ICML)
- Dataset Distillation with Infinitely Wide Convolutional Networks (NIPS)

2022
- Dataset Condensation with Distribution Matching (ICLR)
- Dataset Condensation via Efficient Synthetic-Data Parameterization (ICML)

2022
- CAFÉ: Learning to Condense Dataset by Aligning Features (CVPR)
- Dataset Distillation by Matching Training Trajectories (CVPR oral)

2) Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, Alexei A. Efros. Dataset Distillaion. In arXiv preprint 2018.

Dataset Distillation[2)]

- History

Dataset Distillation, 2018, arXiv

2020

Flexible Dataset Distillation: Learn Labels instead of Images (NIPS Workshop)

Dataset Condensation with Gradient Matching (ICLR)

2021

Dataset Condensation with Differentiable Siamese Augmentation (ICML)

Dataset Distillation with Infinitely Wide Convolutional Networks (NIPS)

2022

Dataset Condensation with Distribution Matching (ICLR)

Dataset Condensation via Efficient Synthetic-Data Parameterization (ICML)

2022

CAFÉ: Learning to Condense Dataset by Aligning Features (CVPR)

Dataset Distillation by Matching Training Trajectories (CVPR oral)

Matching Training Trajectories[3)]

- New formulation that optimizes our distilled data
- Train the network for several iterations on our distilled data and optimize the distilled data
- Outperform existing methods & allow us to distill higher-resolution visual data

3) George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, Jun-Yan Zhu. Dataset Distillation by Matching Training Trajectories. In CVPR 2022.

Matching Training Trajectories[3)]

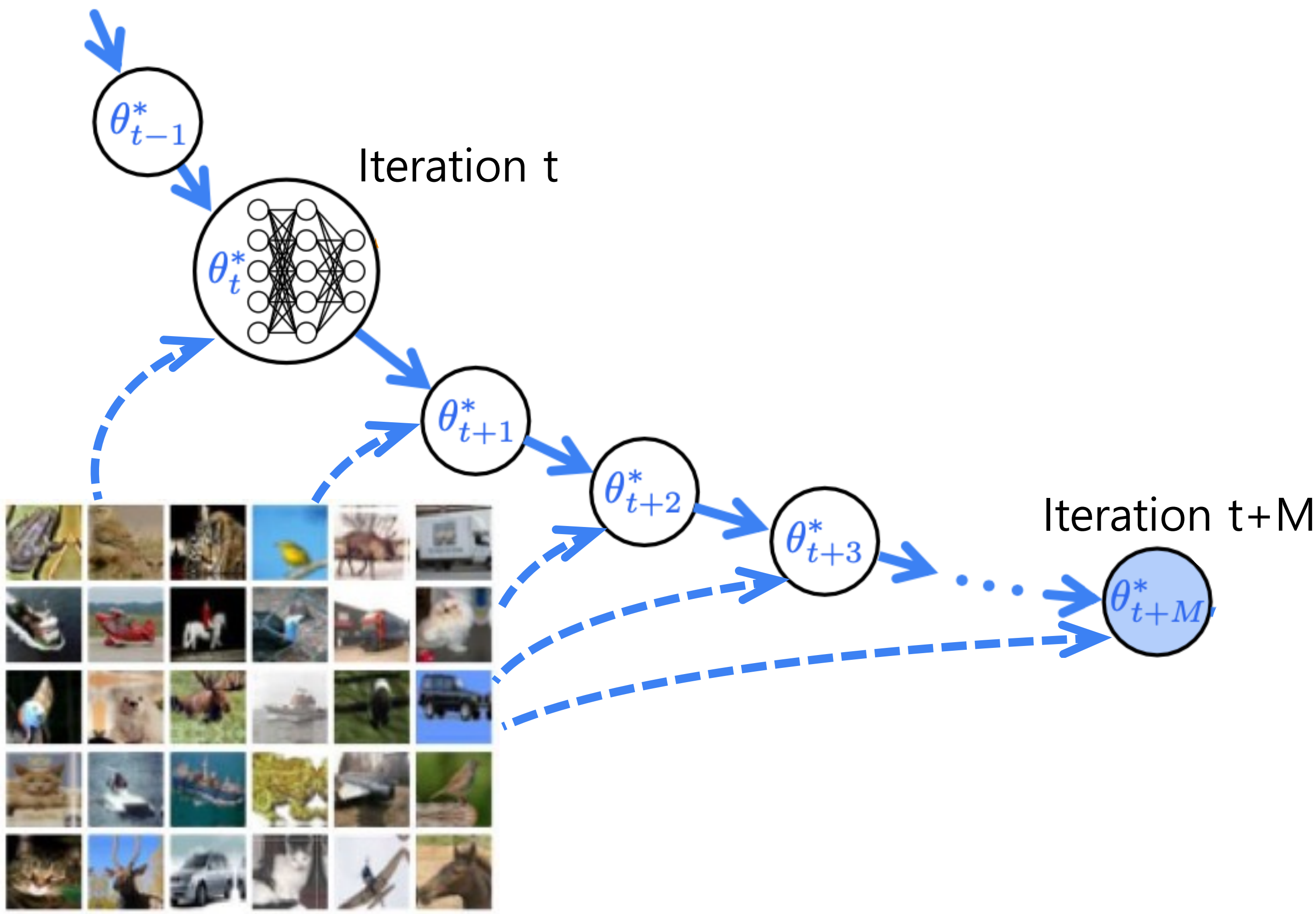
Algorithm 1 Dataset Distillation via Trajectory Matching

Input: $\{\tau_i^*\}$: set of expert parameter trajectories trained on $\mathcal{D}_{\text{real}}$.
Input: M: # of updates between starting and target expert params.
Input: N: # of updates to student network per distillation step.
Input: $\mathcal{A}$: Differentiable augmentation function.
Input: $T^+ < T$: Maximum start epoch.

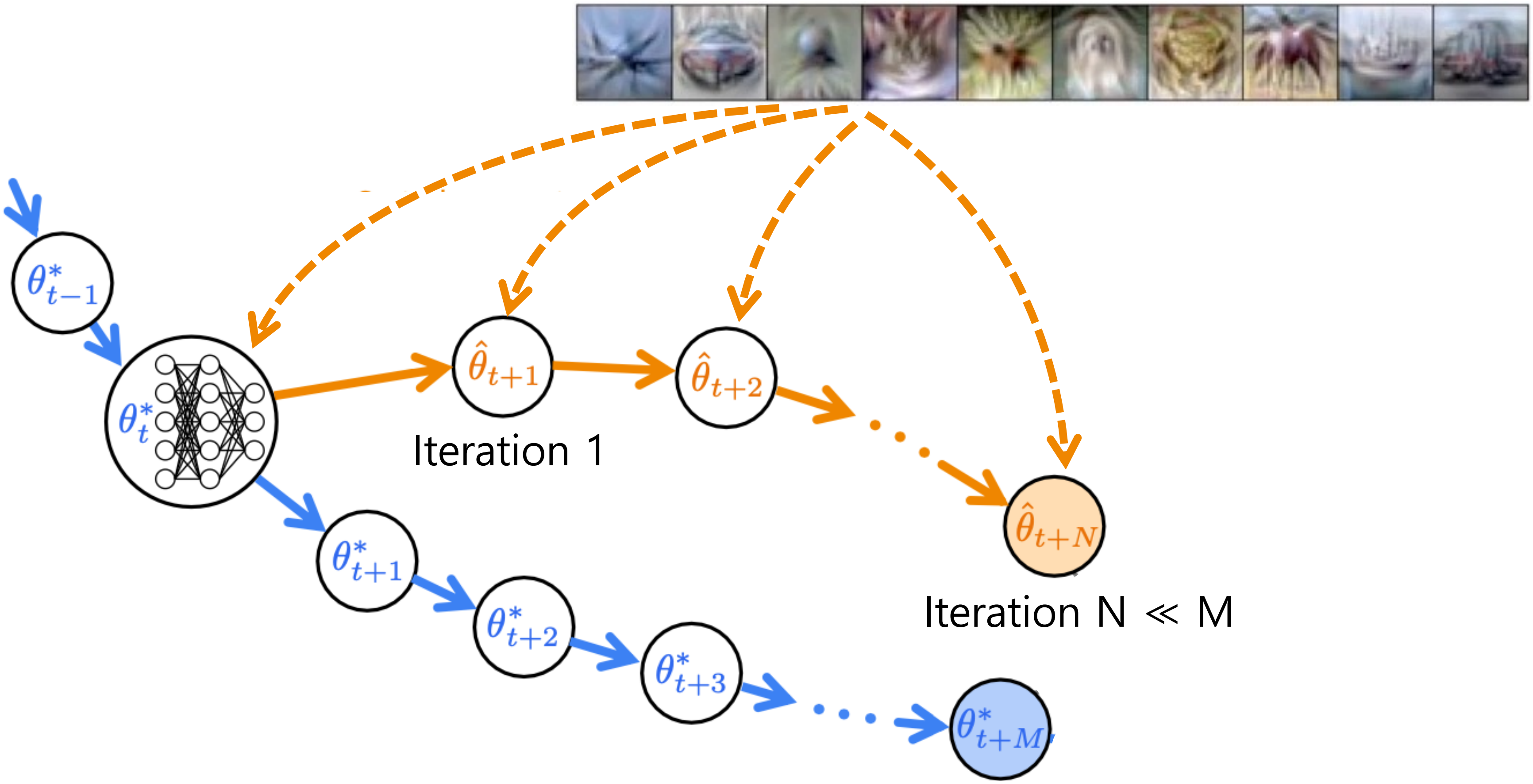
Initialize distilled data $\mathcal{D}_{\text{syn}} \sim \mathcal{D}_{\text{real}}$
Initialize trainable learning rate $\alpha := \alpha_0$ for apply $\mathcal{D}_{\text{syn}}$
for each distillation step... **do**
  ▷ Sample expert trajectory: $\tau^* \sim \{\tau_i^*\}$ with $\tau^* = \{\theta_t^*\}_0^T$
  ▷ Choose random start epoch, $t \leq T^+$
  ▷ Initialize student network with expert params:
    $\hat{\theta}_t := \theta_t^*$
  for $n = 0 \to N-1$ **do**
    ▷ Sample a mini-batch of distilled images:
      $b_{t+n} \sim \mathcal{D}_{\text{syn}}$
    ▷ Update student network w.r.t. classification loss:
      $\hat{\theta}_{t+n+1} = \hat{\theta}_{t+n} - \alpha \nabla \ell(\mathcal{A}(b_{t+n}); \hat{\theta}_{t+n})$
  end for
  ▷ Compute loss between ending student and expert params:
    $\mathcal{L} = \|\hat{\theta}_{t+N} - \theta_{t+M}^*\|_2^2 \;/\; \|\theta_t^* - \theta_{t+M}^*\|_2^2$
  ▷ Update $\mathcal{D}_{\text{syn}}$ and α with respect to $\mathcal{L}$
end for
Output: distilled data $\mathcal{D}_{\text{syn}}$ and learning rate α

Matching Training Trajectories[3)]

Expert Trajectories are trained on Real Data

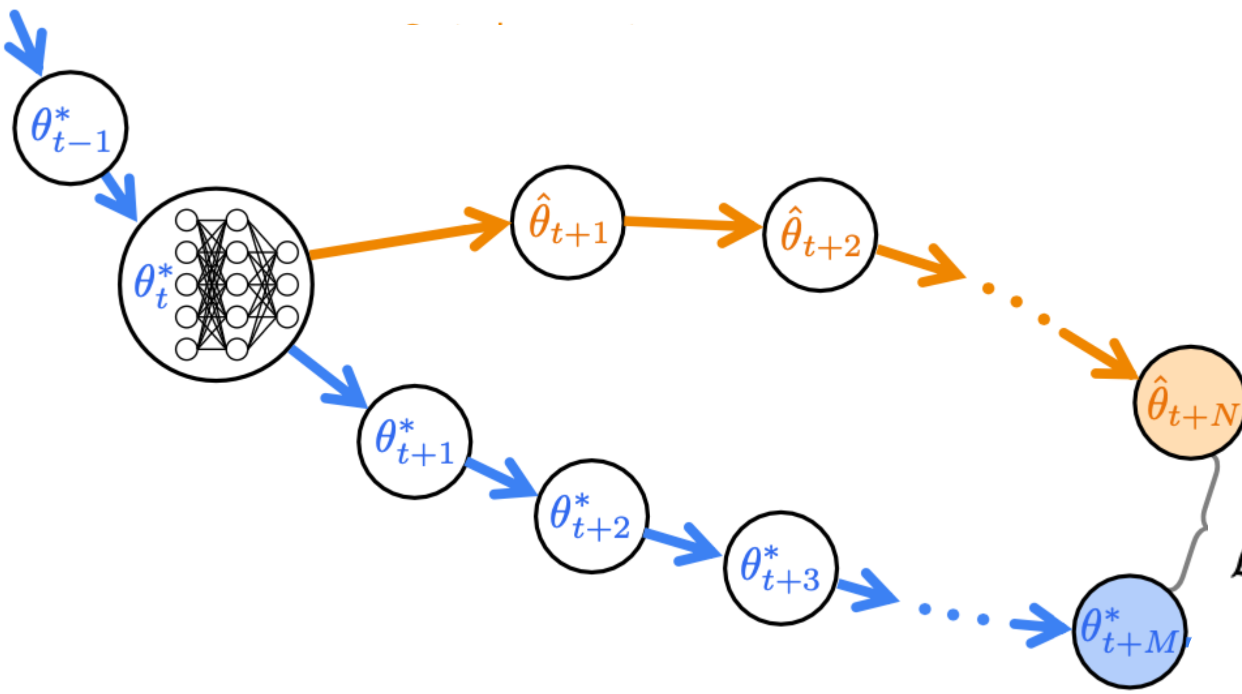


Matching Training Trajectories[3)]



Student Trajectories are trained on Synthetic Data

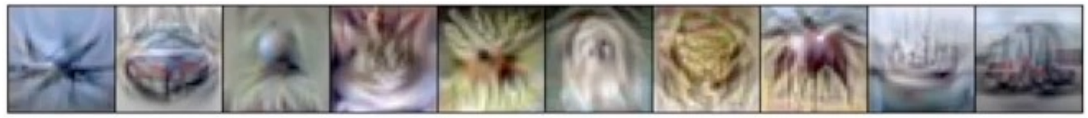
Matching Training Trajectories[3)]



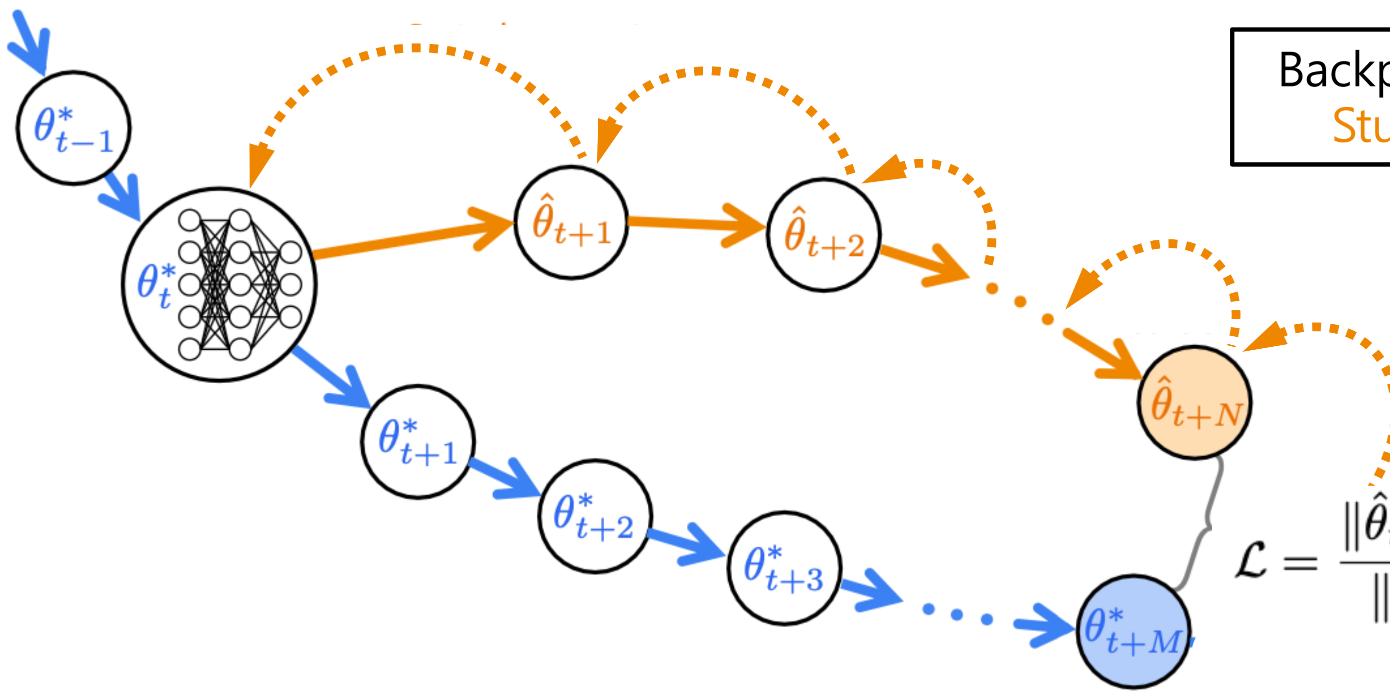
$$\mathcal{L} = \frac{\|\boxed{\hat{\theta}_{t+N}} - \boxed{\theta^*_{t+M}}\|_2^2}{\|\theta^*_t - \theta^*_{t+M}\|_2^2}$$

Relative error between ends of Student and Expert Trajectories

Matching Training Trajectories[3)]

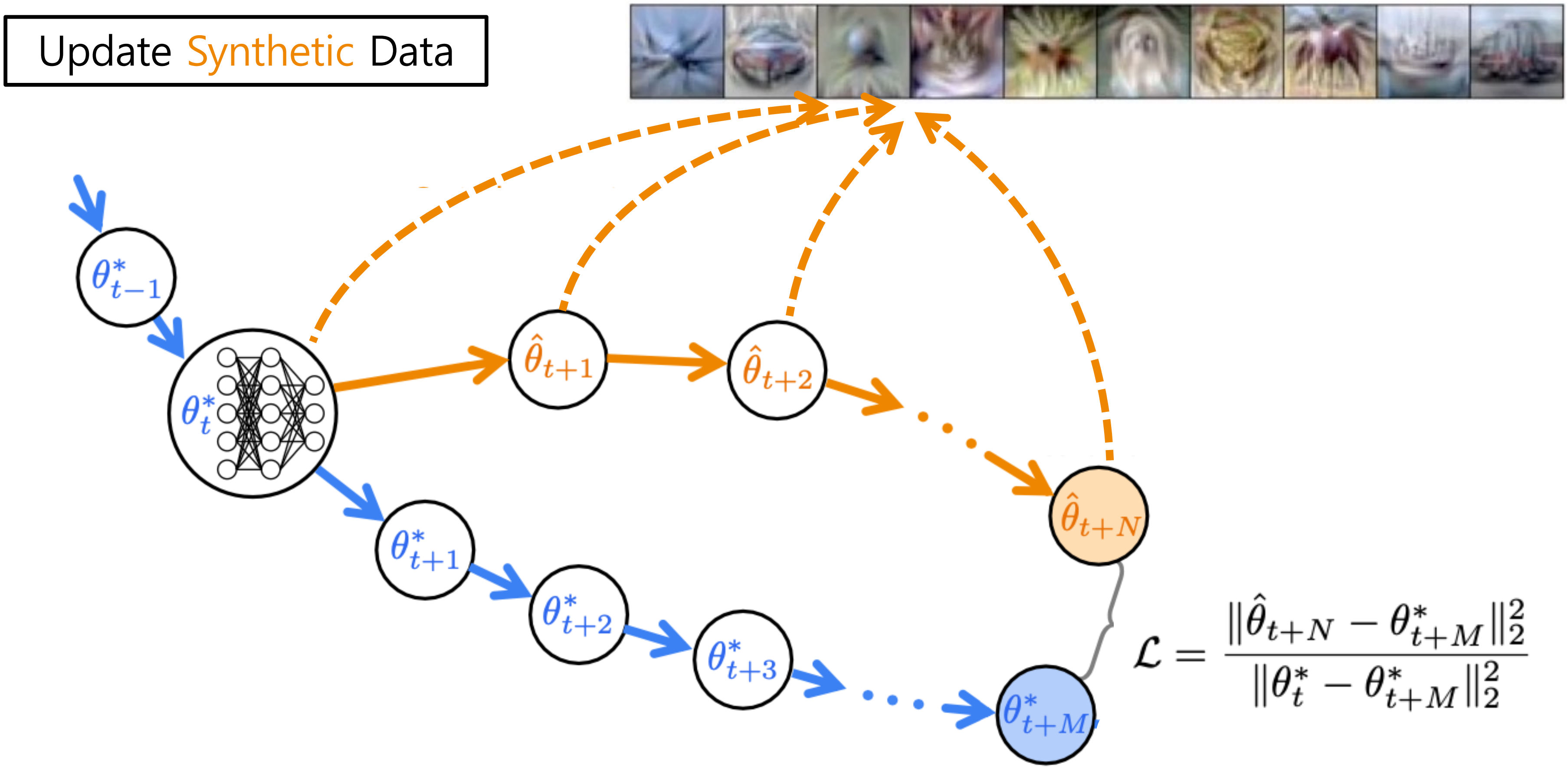


Backprop Through All Student Updates



$$\mathcal{L} = \frac{\|\hat{\theta}_{t+N} - \theta^*_{t+M}\|_2^2}{\|\theta^*_t - \theta^*_{t+M}\|_2^2}$$

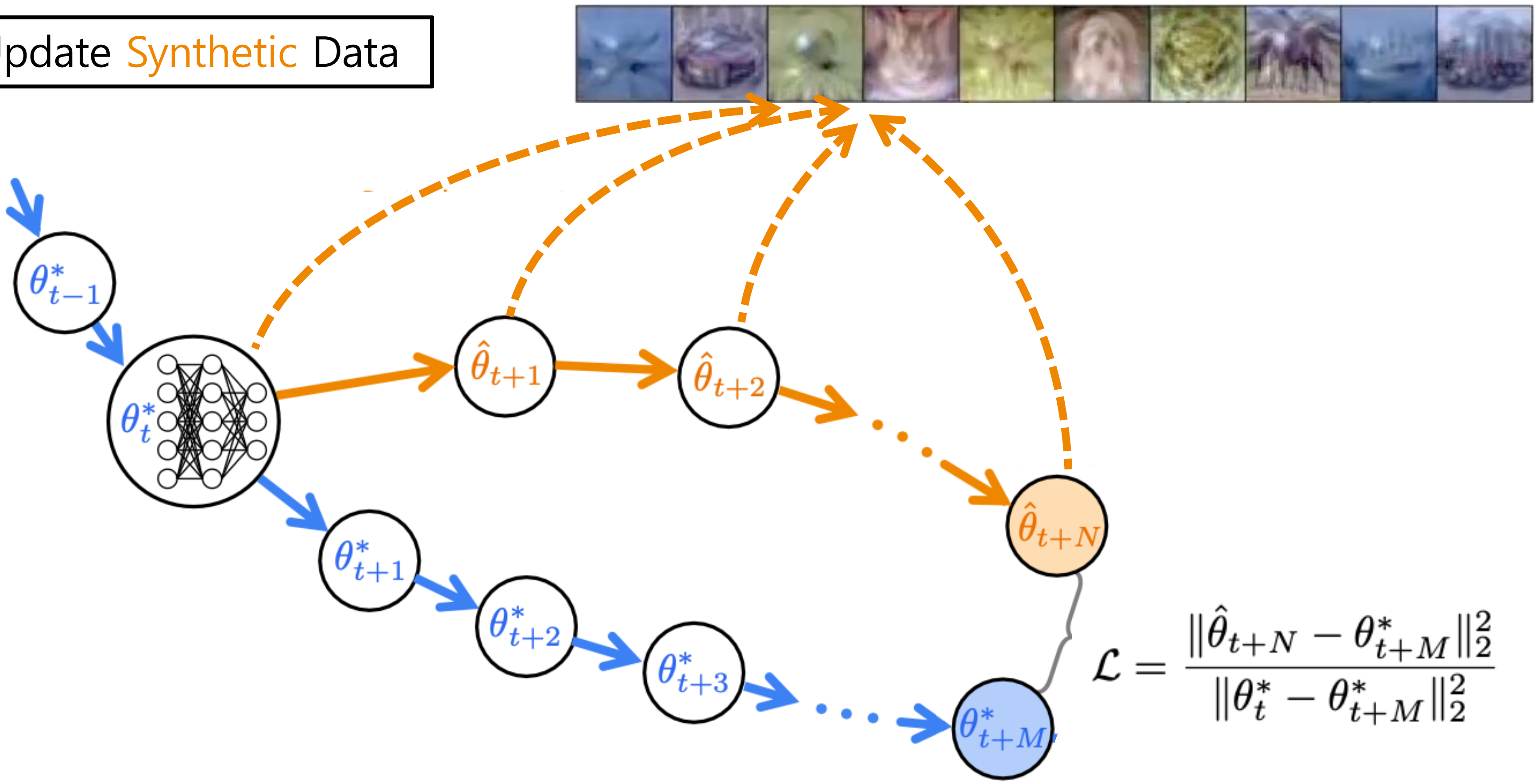
Matching Training Trajectories[3)]



$$\mathcal{L} = \frac{\|\hat{\theta}_{t+N} - \theta^*_{t+M}\|_2^2}{\|\theta^*_t - \theta^*_{t+M}\|_2^2}$$

Matching Training Trajectories[3)]

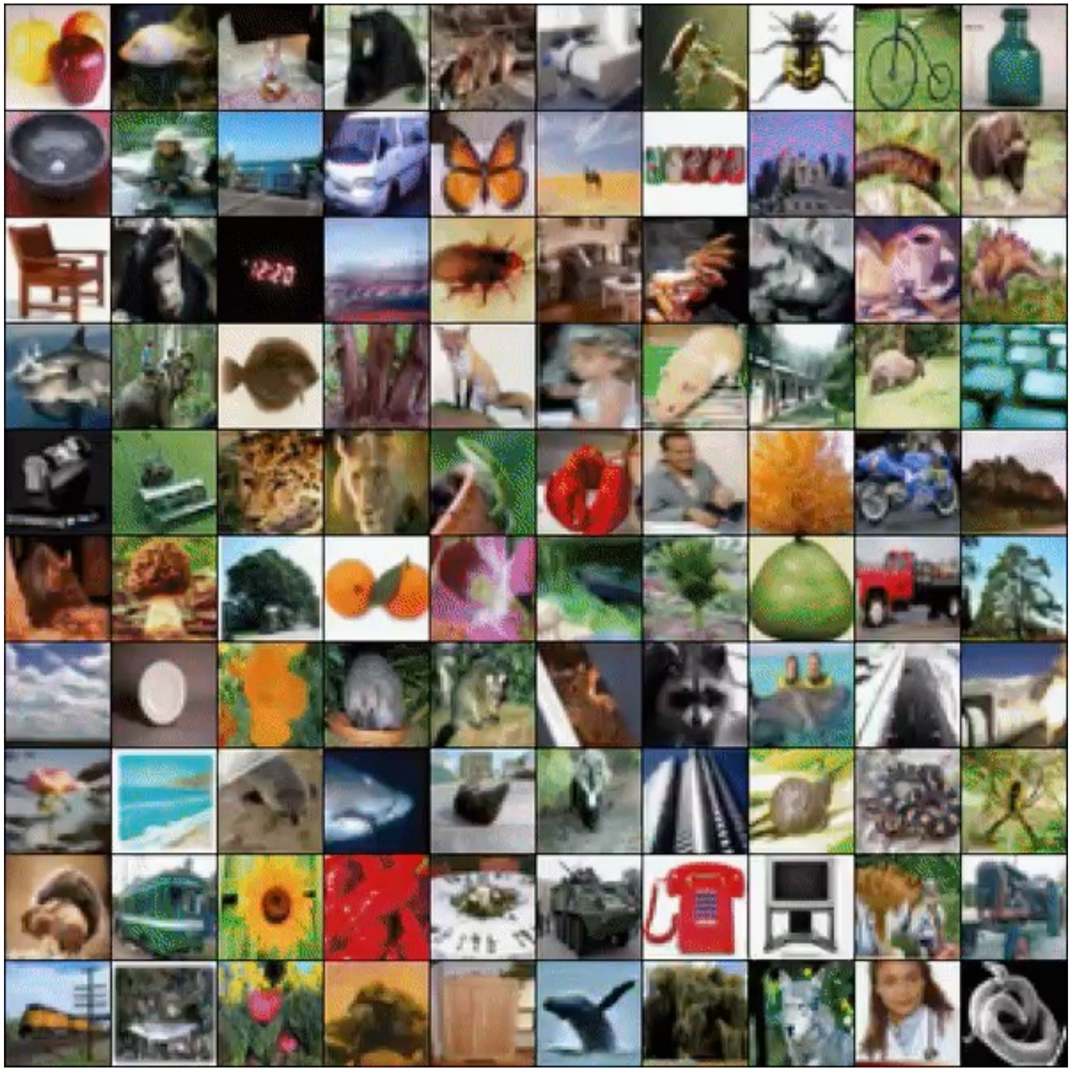
Update Synthetic Data



$$\mathcal{L} = \frac{\|\hat{\theta}_{t+N} - \theta^*_{t+M}\|_2^2}{\|\theta^*_t - \theta^*_{t+M}\|_2^2}$$

Matching Training Trajectories[3)]

1000 distillation iterations of CIFAR-100, 1 image/class



Matching Training Trajectories[3)]

- Experiments
  - 32×32 CIFAR-10 and CIFAR-100
  - 64×64 Tiny ImageNet

	Img/Cls	Ratio %	DD† [44]	LD† [2]	DC [47]	DSA [45]	DM [46]	CAFE [43]	CAFE+DSA [43]	Ours	Full Dataset
			Training Set Synthesis								
CIFAR-10	1	0.02	-	25.7 ± 0.7	28.3 ± 0.5	28.8 ± 0.7	26.0 ± 0.8	30.3 ± 1.1	31.6 ± 0.8	$\mathbf{46.3 \pm 0.8}^*$	84.8 ± 0.1
	10	0.2	36.8 ± 1.2	38.3 ± 0.4	44.9 ± 0.5	52.1 ± 0.5	48.9 ± 0.6	46.3 ± 0.6	50.9 ± 0.5	$\mathbf{65.3 \pm 0.7}^*$	
	50	1	-	42.5 ± 0.4	53.9 ± 0.5	60.6 ± 0.5	63.0 ± 0.4	55.5 ± 0.6	62.3 ± 0.4	$\mathbf{71.6 \pm 0.2}$	
CIFAR-100	1	0.2	-	11.5 ± 0.4	12.8 ± 0.3	13.9 ± 0.3	11.4 ± 0.3	12.9 ± 0.3	14.0 ± 0.3	$\mathbf{24.3 \pm 0.3}^*$	56.2 ± 0.3
	10	2	-	-	25.2 ± 0.3	32.3 ± 0.3	29.7 ± 0.3	27.8 ± 0.3	31.5 ± 0.2	$\mathbf{40.1 \pm 0.4}$	
	50	10	-	-	-	42.8 ± 0.4	43.6 ± 0.4	37.9 ± 0.3	42.9 ± 0.2	$\mathbf{47.7 \pm 0.2}^*$	
Tiny ImageNet	1	0.2	-	-	-	-	3.9 ± 0.2	-	-	$\mathbf{8.8 \pm 0.3}$	37.6 ± 0.4
	10	2	-	-	-	-	12.9 ± 0.4	-	-	$\mathbf{23.2 \pm 0.2}$	
	50	10	-	-	-	-	24.1 ± 0.3	-	-	$\mathbf{28.0 \pm 0.3}$	

  - 128×128 ImageNet subsets

	ImageNette	ImageWoof	ImageFruit	ImageMeow	ImageSquawk	ImageYellow
1 Img/Cls	47.7 ± 0.9	28.6 ± 0.8	26.6 ± 0.8	30.7 ± 1.6	39.4 ± 1.5	45.2 ± 0.8
10 Img/Cls	63.0 ± 1.3	35.8 ± 1.8	40.3 ± 1.3	40.4 ± 2.2	52.3 ± 1.0	60.0 ± 1.5
Full Dataset	87.4 ± 1.0	67.0 ± 1.3	63.9 ± 2.0	66.7 ± 1.1	87.5 ± 0.3	84.4 ± 0.6

Matching Training Trajectories[3)]

- Experiments
  - 32×32 CIFAR-10 and CIFAR-100
  - 64×64 Tiny ImageNet

	Img/Cls	Ratio %	Training Set Synthesis DD† [44]	LD† [2]	DC [47]	DSA [45]	DM [46]	CAFE [43]	CAFE+DSA [43]	Ours	Full Dataset
CIFAR-10	1	0.02	-	25.7 ± 0.7	28.3 ± 0.5	28.8 ± 0.7	26.0 ± 0.8	30.3 ± 1.1	31.6 ± 0.8	**46.3 ± 0.8***	84.8 ± 0.1
	10	0.2	36.8 ± 1.2	38.3 ± 0.4	44.9 ± 0.5	52.1 ± 0.5	48.9 ± 0.6	46.3 ± 0.6	50.9 ± 0.5	**65.3 ± 0.7***	
	50	1	-	42.5 ± 0.4	53.9 ± 0.5	60.6 ± 0.5	63.0 ± 0.4	55.5 ± 0.6	62.3 ± 0.4	**71.6 ± 0.2**	
CIFAR-100	1	0.2	-	11.5 ± 0.4	12.8 ± 0.3	13.9 ± 0.3	11.4 ± 0.3	12.9 ± 0.3	14.0 ± 0.3	**24.3 ± 0.3***	56.2 ± 0.3
	10	2	-	-	25.2 ± 0.3	32.3 ± 0.3	29.7 ± 0.3	27.8 ± 0.3	31.5 ± 0.2	**40.1 ± 0.4**	
	50	10	-	-	-	42.8 ± 0.4	43.6 ± 0.4	37.9 ± 0.3	42.9 ± 0.2	**47.7 ± 0.2***	
Tiny ImageNet	1	0.2	-	-	-	-	3.9 ± 0.2	-	-	**8.8 ± 0.3**	37.6 ± 0.4
	10	2	-	-	-	-	12.9 ± 0.4	-	-	**23.2 ± 0.2**	
	50	10	-	-	-	-	24.1 ± 0.3	-	-	**28.0 ± 0.3**	

  - 128×128 ImageNet subsets

	ImageNette	ImageWoof	ImageFruit	ImageMeow	ImageSquawk	ImageYellow
1 Img/Cls	47.7 ± 0.9	28.6 ± 0.8	26.6 ± 0.8	30.7 ±1.6	39.4 ± 1.5	45.2 ± 0.8
10 Img/Cls	63.0 ± 1.3	35.8 ± 1.8	40.3 ± 1.3	40.4 ± 2.2	52.3 ± 1.0	60.0 ± 1.5
Full Dataset	87.4 ± 1.0	67.0 ± 1.3	63.9 ± 2.0	66.7 ± 1.1	87.5 ± 0.3	84.4 ± 0.6

Matching Training Trajectories[3)]

50 images/class

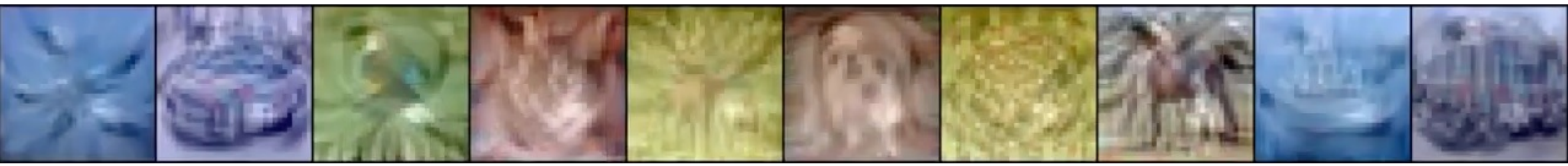
Matching Training Trajectories[3)]

10 images/class

Matching Training Trajectories[3)]

1 image/class

Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Boat	Truck



Matching Training Trajectories[3)]

- Experiments
  - 32×32 CIFAR-10 and CIFAR-100
  - 64×64 Tiny ImageNet

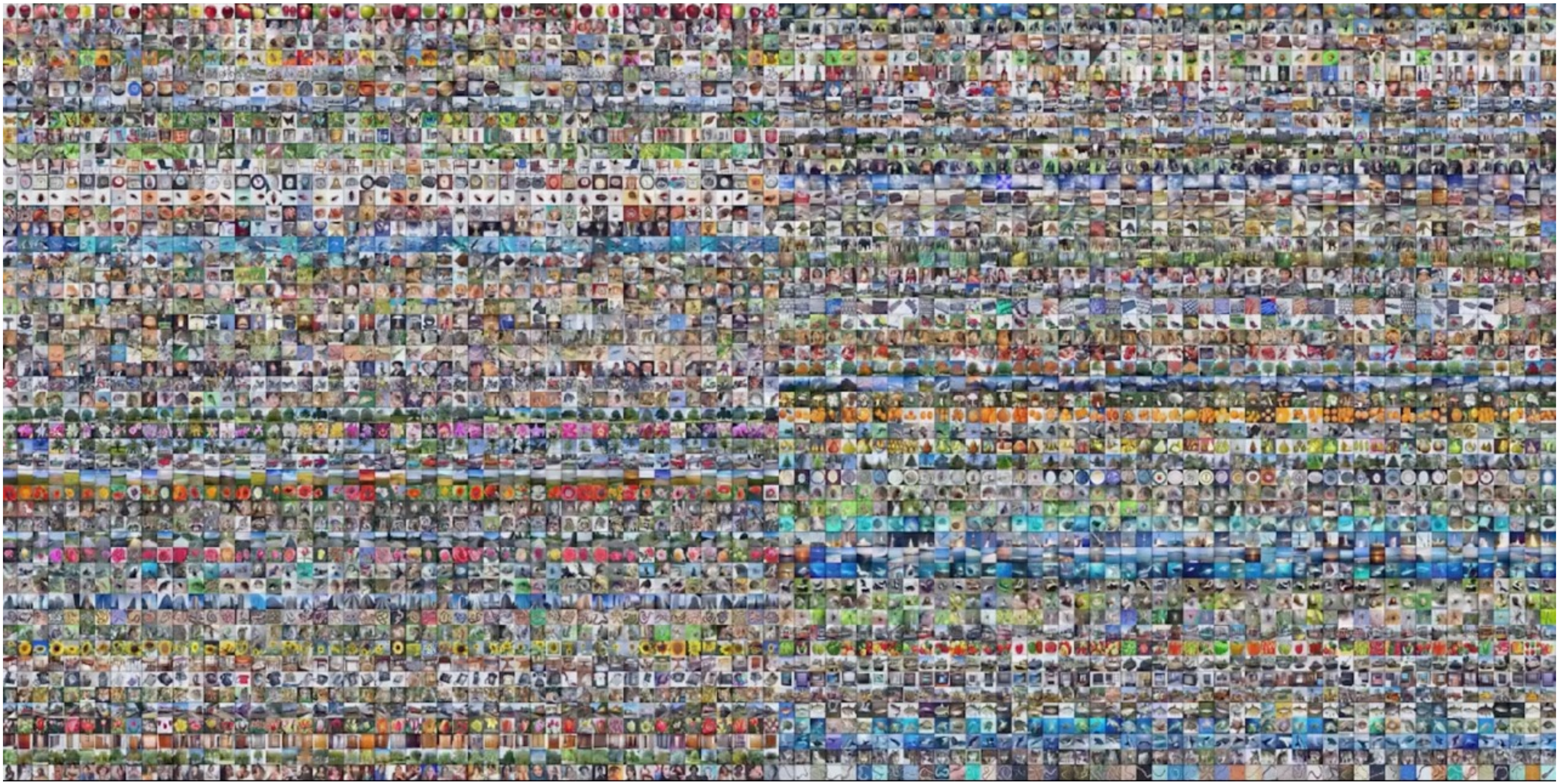
	Img/Cls	Ratio %	DD† [44]	LD† [2]	DC [47]	DSA [45]	DM [46]	CAFE [43]	CAFE+DSA [43]	Ours	Full Dataset
			Training Set Synthesis								
CIFAR-10	1	0.02	-	25.7 ± 0.7	28.3 ± 0.5	28.8 ± 0.7	26.0 ± 0.8	30.3 ± 1.1	31.6 ± 0.8	**46.3 ± 0.8***	84.8 ± 0.1
	10	0.2	36.8 ± 1.2	38.3 ± 0.4	44.9 ± 0.5	52.1 ± 0.5	48.9 ± 0.6	46.3 ± 0.6	50.9 ± 0.5	**65.3 ± 0.7***	
	50	1	-	42.5 ± 0.4	53.9 ± 0.5	60.6 ± 0.5	63.0 ± 0.4	55.5 ± 0.6	62.3 ± 0.4	**71.6 ± 0.2**	
CIFAR-100	1	0.2	-	11.5 ± 0.4	12.8 ± 0.3	13.9 ± 0.3	11.4 ± 0.3	12.9 ± 0.3	14.0 ± 0.3	**24.3 ± 0.3***	56.2 ± 0.3
	10	2	-	-	25.2 ± 0.3	32.3 ± 0.3	29.7 ± 0.3	27.8 ± 0.3	31.5 ± 0.2	**40.1 ± 0.4**	
	50	10	-	-	-	42.8 ± 0.4	43.6 ± 0.4	37.9 ± 0.3	42.9 ± 0.2	**47.7 ± 0.2***	
Tiny ImageNet	1	0.2	-	-	-	-	3.9 ± 0.2	-	-	**8.8 ± 0.3**	37.6 ± 0.4
	10	2	-	-	-	-	12.9 ± 0.4	-	-	**23.2 ± 0.2**	
	50	10	-	-	-	-	24.1 ± 0.3	-	-	**28.0 ± 0.3**	

  - 128×128 ImageNet subsets

	ImageNette	ImageWoof	ImageFruit	ImageMeow	ImageSquawk	ImageYellow
1 Img/Cls	47.7 ± 0.9	28.6 ± 0.8	26.6 ± 0.8	30.7 ±1.6	39.4 ± 1.5	45.2 ± 0.8
10 Img/Cls	63.0 ± 1.3	35.8 ± 1.8	40.3 ± 1.3	40.4 ± 2.2	52.3 ± 1.0	60.0 ± 1.5
Full Dataset	87.4 ± 1.0	67.0 ± 1.3	63.9 ± 2.0	66.7 ± 1.1	87.5 ± 0.3	84.4 ± 0.6

Matching Training Trajectories[3)]

50 images/class



Matching Training Trajectories[3)]

10 images/class

Matching Training Trajectories[3)]

1 image/class

Matching Training Trajectories[3)]

- Experiments
  - 32×32 CIFAR-10 and CIFAR-100
  - 64×64 Tiny ImageNet

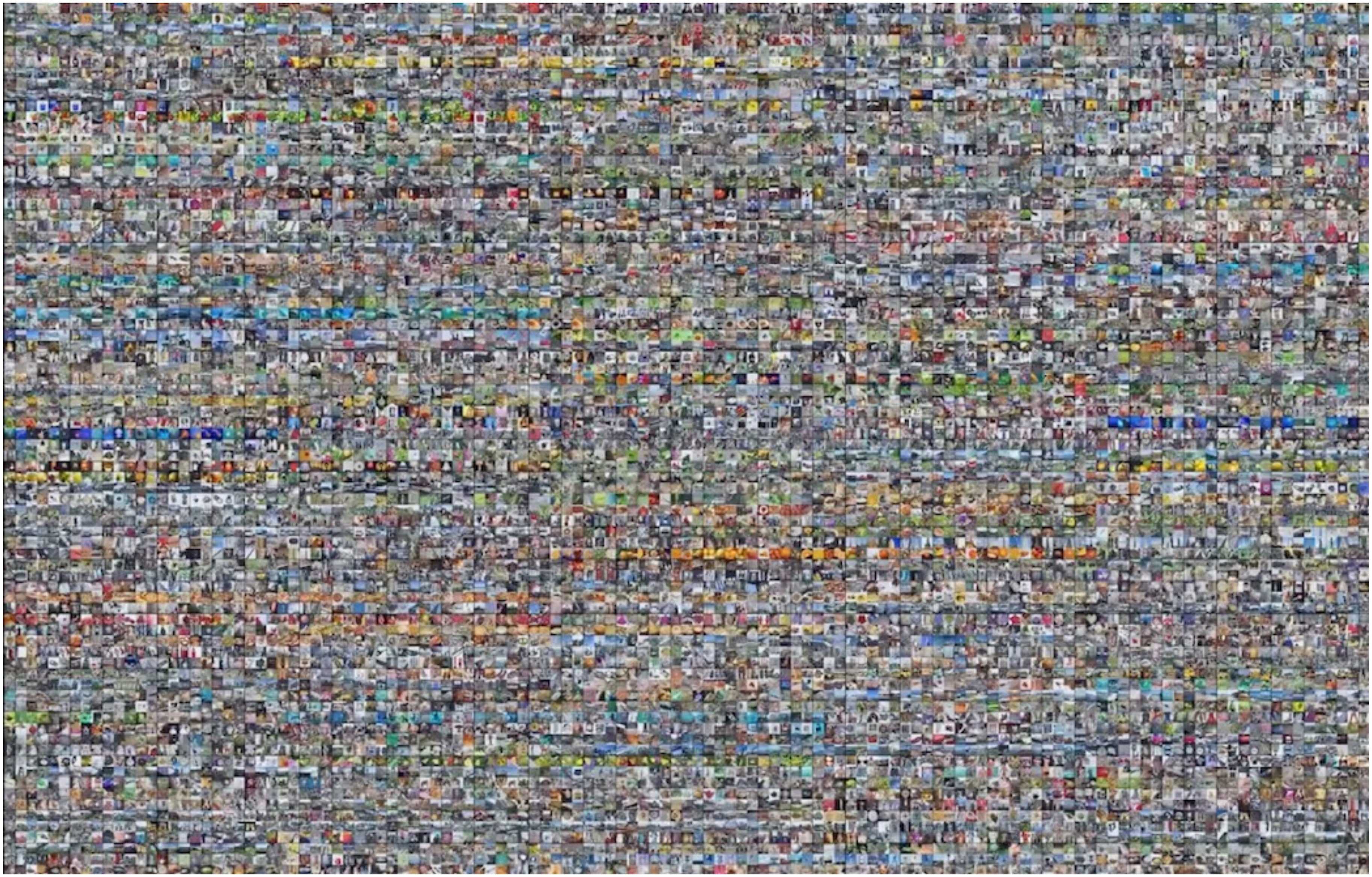
	Img/Cls	Ratio %	DD† [44]	LD† [2]	DC [47]	DSA [45]	DM [46]	CAFE [43]	CAFE+DSA [43]	Ours	Full Dataset
			Training Set Synthesis								
CIFAR-10	1	0.02	-	25.7 ± 0.7	28.3 ± 0.5	28.8 ± 0.7	26.0 ± 0.8	30.3 ± 1.1	31.6 ± 0.8	**46.3 ± 0.8***	84.8 ± 0.1
	10	0.2	36.8 ± 1.2	38.3 ± 0.4	44.9 ± 0.5	52.1 ± 0.5	48.9 ± 0.6	46.3 ± 0.6	50.9 ± 0.5	**65.3 ± 0.7***	
	50	1	-	42.5 ± 0.4	53.9 ± 0.5	60.6 ± 0.5	63.0 ± 0.4	55.5 ± 0.6	62.3 ± 0.4	**71.6 ± 0.2**	
CIFAR-100	1	0.2	-	11.5 ± 0.4	12.8 ± 0.3	13.9 ± 0.3	11.4 ± 0.3	12.9 ± 0.3	14.0 ± 0.3	**24.3 ± 0.3***	56.2 ± 0.3
	10	2	-	-	25.2 ± 0.3	32.3 ± 0.3	29.7 ± 0.3	27.8 ± 0.3	31.5 ± 0.2	**40.1 ± 0.4**	
	50	10	-	-	-	42.8 ± 0.4	43.6 ± 0.4	37.9 ± 0.3	42.9 ± 0.2	**47.7 ± 0.2***	
Tiny ImageNet	1	0.2	-	-	-	-	3.9 ± 0.2	-	-	**8.8 ± 0.3**	37.6 ± 0.4
	10	2	-	-	-	-	12.9 ± 0.4	-	-	**23.2 ± 0.2**	
	50	10	-	-	-	-	24.1 ± 0.3	-	-	**28.0 ± 0.3**	

  - 128×128 ImageNet subsets

	ImageNette	ImageWoof	ImageFruit	ImageMeow	ImageSquawk	ImageYellow
1 Img/Cls	47.7 ± 0.9	28.6 ± 0.8	26.6 ± 0.8	30.7 ±1.6	39.4 ± 1.5	45.2 ± 0.8
10 Img/Cls	63.0 ± 1.3	35.8 ± 1.8	40.3 ± 1.3	40.4 ± 2.2	52.3 ± 1.0	60.0 ± 1.5
Full Dataset	87.4 ± 1.0	67.0 ± 1.3	63.9 ± 2.0	66.7 ± 1.1	87.5 ± 0.3	84.4 ± 0.6

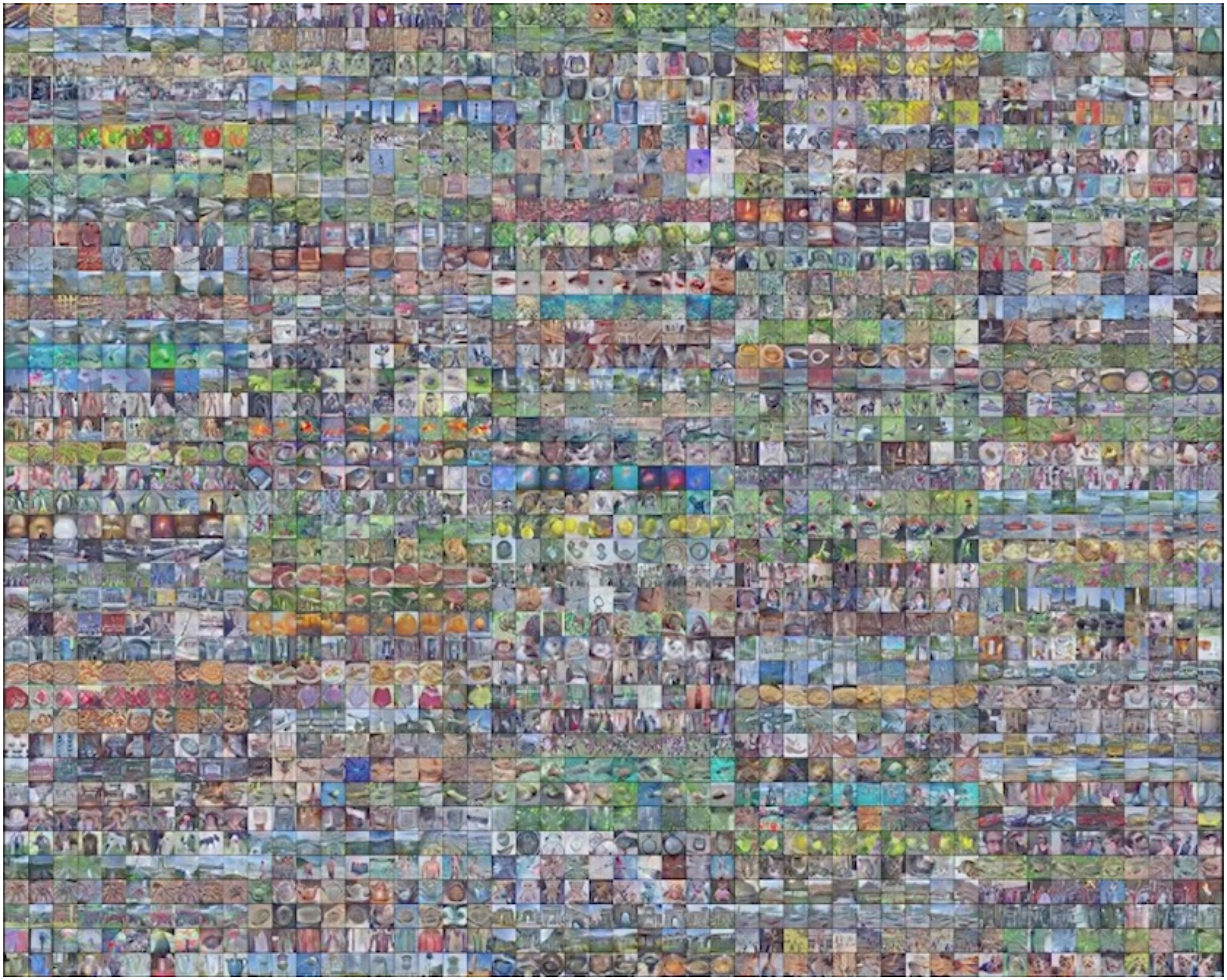
Matching Training Trajectories[3)]

50 images/class



Matching Training Trajectories[3)]

10 images/class



Matching Training Trajectories[3)]

1 image/class

Matching Training Trajectories[3)]

- Experiments
  - 32×32 CIFAR-10 and CIFAR-100
  - 64×64 Tiny ImageNet

	Img/Cls	Ratio %	DD† [44]	LD† [2]	DC [47]	DSA [45]	DM [46]	CAFE [43]	CAFE+DSA [43]	Ours	Full Dataset
			Training Set Synthesis								
CIFAR-10	1	0.02	-	25.7 ± 0.7	28.3 ± 0.5	28.8 ± 0.7	26.0 ± 0.8	30.3 ± 1.1	31.6 ± 0.8	**46.3 ± 0.8***	84.8 ± 0.1
	10	0.2	36.8 ± 1.2	38.3 ± 0.4	44.9 ± 0.5	52.1 ± 0.5	48.9 ± 0.6	46.3 ± 0.6	50.9 ± 0.5	**65.3 ± 0.7***	
	50	1	-	42.5 ± 0.4	53.9 ± 0.5	60.6 ± 0.5	63.0 ± 0.4	55.5 ± 0.6	62.3 ± 0.4	**71.6 ± 0.2**	
CIFAR-100	1	0.2	-	11.5 ± 0.4	12.8 ± 0.3	13.9 ± 0.3	11.4 ± 0.3	12.9 ± 0.3	14.0 ± 0.3	**24.3 ± 0.3***	56.2 ± 0.3
	10	2	-	-	25.2 ± 0.3	32.3 ± 0.3	29.7 ± 0.3	27.8 ± 0.3	31.5 ± 0.2	**40.1 ± 0.4**	
	50	10	-	-	-	42.8 ± 0.4	43.6 ± 0.4	37.9 ± 0.3	42.9 ± 0.2	**47.7 ± 0.2***	
Tiny ImageNet	1	0.2	-	-	-	-	3.9 ± 0.2	-	-	**8.8 ± 0.3**	37.6 ± 0.4
	10	2	-	-	-	-	12.9 ± 0.4	-	-	**23.2 ± 0.2**	
	50	10	-	-	-	-	24.1 ± 0.3	-	-	**28.0 ± 0.3**	

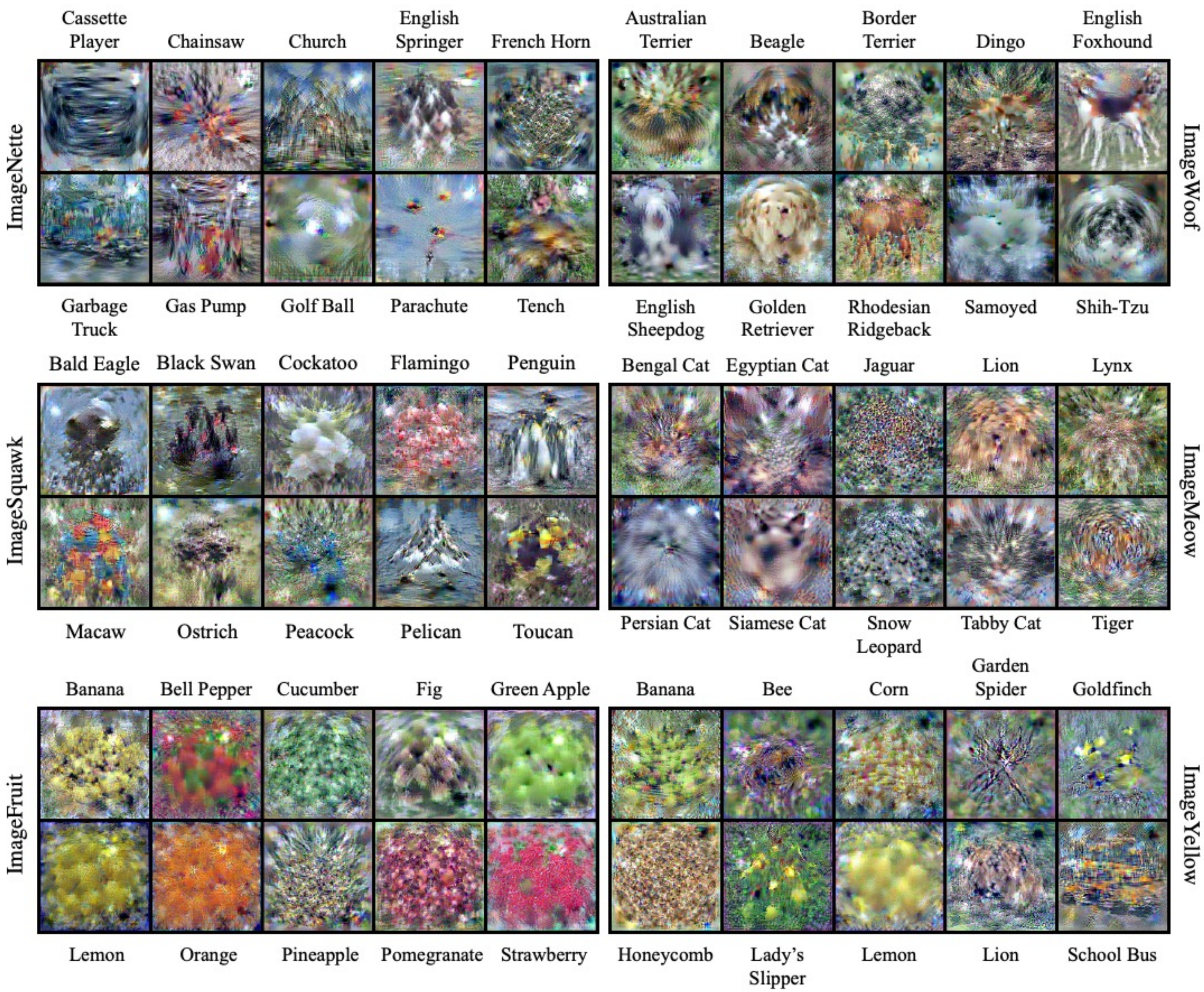
  - 128×128 ImageNet Subsets

	ImageNette	ImageWoof	ImageFruit	ImageMeow	ImageSquawk	ImageYellow
1 Img/Cls	47.7 ± 0.9	28.6 ± 0.8	26.6 ± 0.8	30.7 ±1.6	39.4 ± 1.5	45.2 ± 0.8
10 Img/Cls	63.0 ± 1.3	35.8 ± 1.8	40.3 ± 1.3	40.4 ± 2.2	52.3 ± 1.0	60.0 ± 1.5
Full Dataset	87.4 ± 1.0	67.0 ± 1.3	63.9 ± 2.0	66.7 ± 1.1	87.5 ± 0.3	84.4 ± 0.6

3) George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, Jun-Yan Zhu. Dataset Distillation by Matching Training Trajectories. In CVPR 2022.

Matching Training Trajectories[3)]

1 image/class



서강대학교 SOGANG UNIVERSITY

VDS LAB

Matching Training Trajectories[3)]

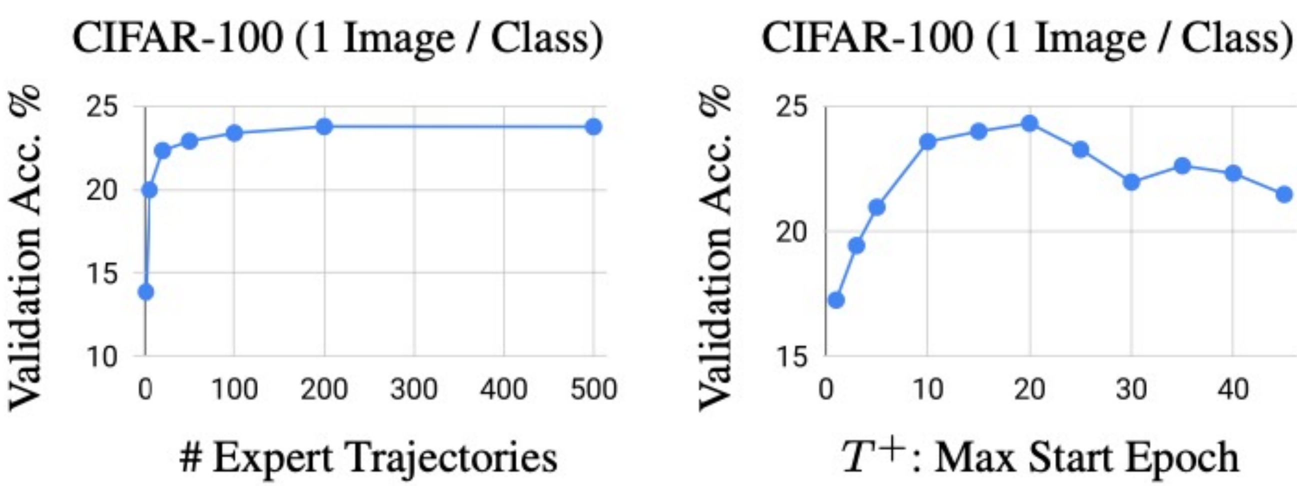
- Experiments
  - Cross-Architecture Generalization
    - Evaluate how well our synthetic data performs on various architectures
    - Robust to changes in architectures
    - Do not seem to suffer from much over-fitting to that model

		Evaluation Model			
		ConvNet	ResNet	VGG	AlexNet
Method	Ours	**64.3 ± 0.7**	**46.4 ± 0.6**	**50.3 ± 0.8**	**34.2 ± 2.6**
	DSA	52.1 ± 0.4	42.8 ± 1.0	43.2 ± 0.5	**35.9 ± 1.3**
	KIP	47.6 ± 0.9	36.8 ± 1.0	42.1 ± 0.4	24.4 ± 3.9

CIFAR-10 with 10 images/class

Matching Training Trajectories[3)]

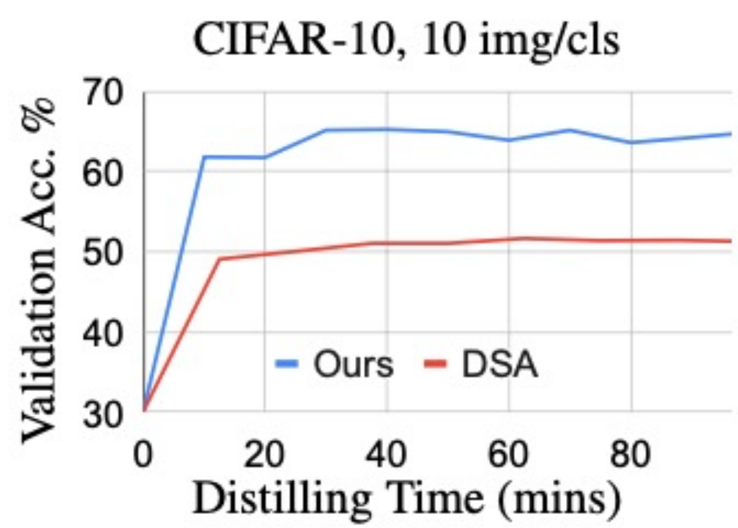
- Experiments
  - Performance w.r.t. the number of expert trajectories (left)
    - Logarithmic performance improvement
    - Quickly saturating near 200
  - Performance w.r.t. expert time-step stage (right)
    - The upper bound on the expert epoch at which the synthetic data starts working cannot be too high or low to ensure quality learning signal.



Matching Training Trajectories[3)]

- Experiments
  - Distillation time
    - 0.6 seconds per distillation step
      - A single RTX3090
      - CIFAR-100, 1 image/class with N = 20

Dataset	Img/Cls	1 Iter. (sec)	1k Iter. (min)	5k Iter. (min)	10k Iter. (min)
CIFAR-10	1	0.5	8	42	83
	10	0.6	10	50	100
	50	0.8	13	67	133
CIFAR-100	1	0.6	10	50	100
	10	0.8	13	67	133
	50	1.9	32	158	317
Tiny ImageNet	1	1.1	18	92	183
	10	2.3	38	192	383
	50	2.6	43	217	433



Conclusion

- Discussion
  - Directly optimizing the synthetic data
    - Induce similar network training dynamics as the real data
  - First to scale to 128×128 ImageNet images
    - Allow us to gain interesting insights of the dataset
    - Serve as an important step towards practical applications of dataset distillation on real-world datasets
- Limitations
  - The computational overhead of training and storing expert trajectories
  - Application to other tasks and datasets with higher resolution