ECCV
EUROPEAN CONFERENCE ON COMPUTER VISION
MILANO 2024

Teddy: Efficient Large-Scale Dataset Distillation via Taylor-Approximated Matching

Ruonan Yu, Songhua Liu, Jingwen Ye, Xinchao Wang*

National University of Singapore, Learning and Vision Lab

NUS National University of Singapore

http://www.lv-nus.org/

Current Challenges in Dataset Distillation

Traditional DD:

- Bi-level optimization:

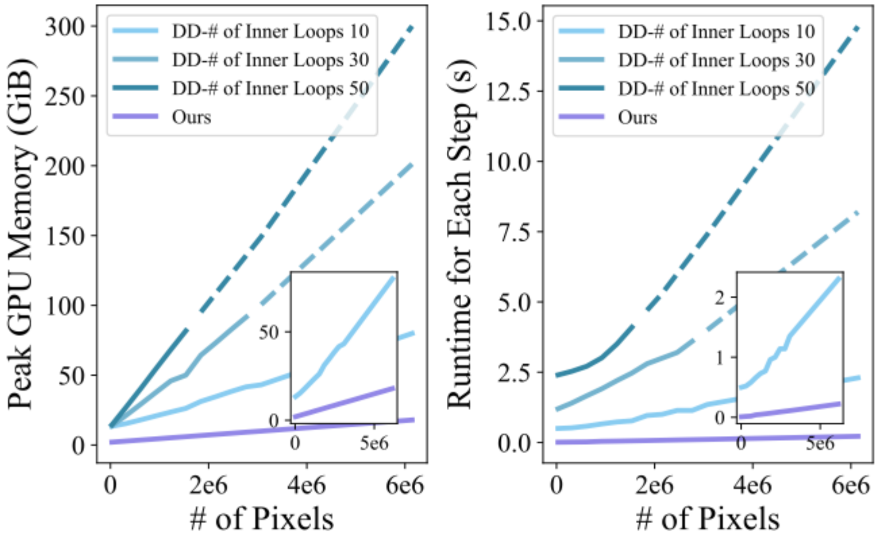
$$\mathcal{L}(\mathcal{S},\mathcal{T}) = \mathbb{E}_{\theta^{(0)}\sim\Theta}[l_{ce}\left(\mathcal{T},\theta_{\mathcal{S}}^{(T)}\right)]$$

$$\theta_{\mathcal{S}}^{(t)} = \theta_{\mathcal{S}}^{(t-1)} - \eta\nabla l_{ce}\left(\mathcal{S};\theta_{\mathcal{S}}^{(t-1)}\right)$$

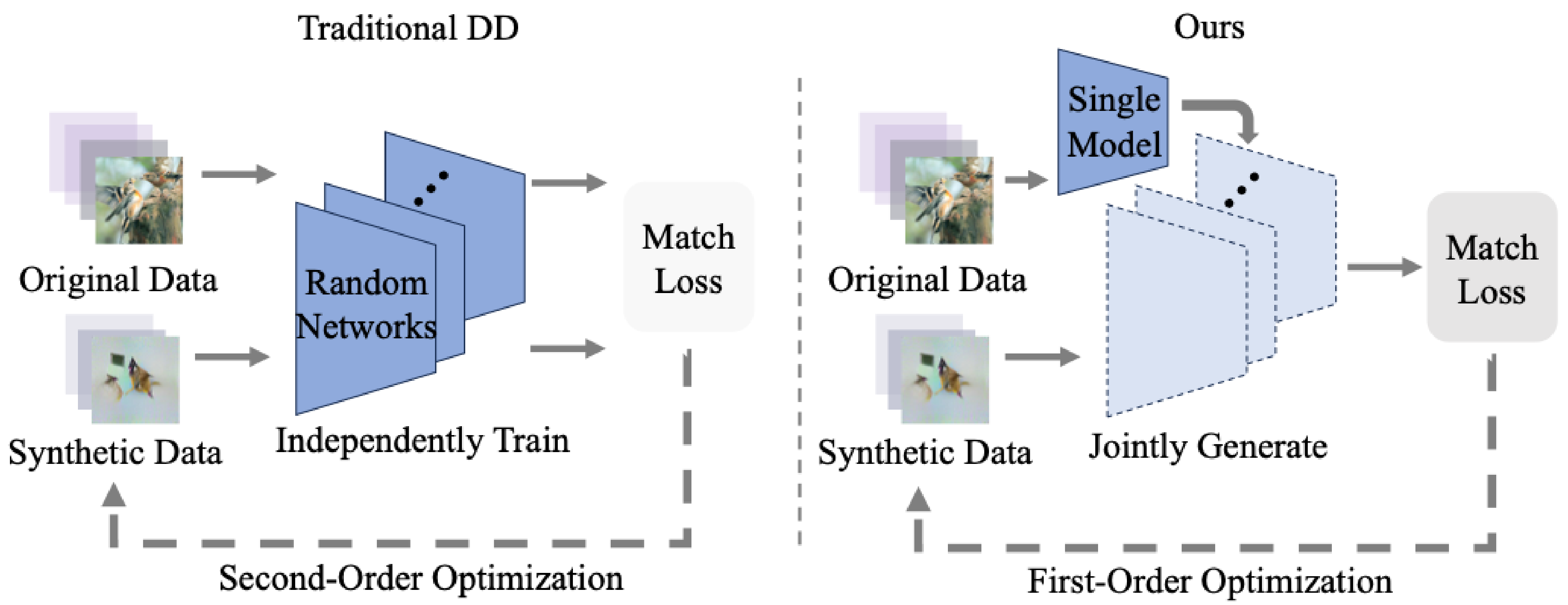
- Re-train a new model for each iteration

High GPU Memory & Time Complexity

Scaling up Problem!



Teddy Efficient Large-Scale Dataset Distillation via Taylor-Approximated Matching



Taylor-Approximated Matching

Start with traditional DD:

$$\mathcal{L}(\mathcal{S},\mathcal{T}) = \mathbb{E}_{\theta^{(0)}\sim\Theta}[l_{ce}\left(\mathcal{T},\theta_{\mathcal{S}}^{(T)}\right)]$$
$$\theta_{\mathcal{S}}^{(t)} = \theta_{\mathcal{S}}^{(t-1)} - \eta\nabla l_{ce}\left(\mathcal{S},\theta_{\mathcal{S}}^{(t-1)}\right)$$

$$l\left(\mathcal{S},\theta_{\mathcal{S}}^{(T)}\right) < \epsilon$$

Using Taylor Expansion:

$$\mathbb{E}_{\theta^{(0)}\sim\Theta} l\left(\mathcal{T},\theta_{\mathcal{S}}^{(T-1)}\right) - \alpha g_{\mathcal{T}}^{(T-1)} \cdot g_{\mathcal{S}}^{(T-1)} = \mathbb{E}_{\theta^{(0)}\sim\Theta} l\left(\mathcal{T},\theta_{\mathcal{S}}^{(0)}\right) - \alpha \sum_{t=0}^{T-1} g_{\mathcal{T}}^{(t)} \cdot g_{\mathcal{S}}^{(t)}$$

Transformed into the sum of the gradient matching of the distilled data and original data

$$g \xrightarrow{\theta^{(0)}\sim\Theta} \frac{1}{|X|}\sigma^2\left(f_\theta(X)\right)W - \frac{1}{|X|}\mu(f_\theta(X))$$

In feature space, gradient matching is equivalent to first-order and second-order statistic information matching

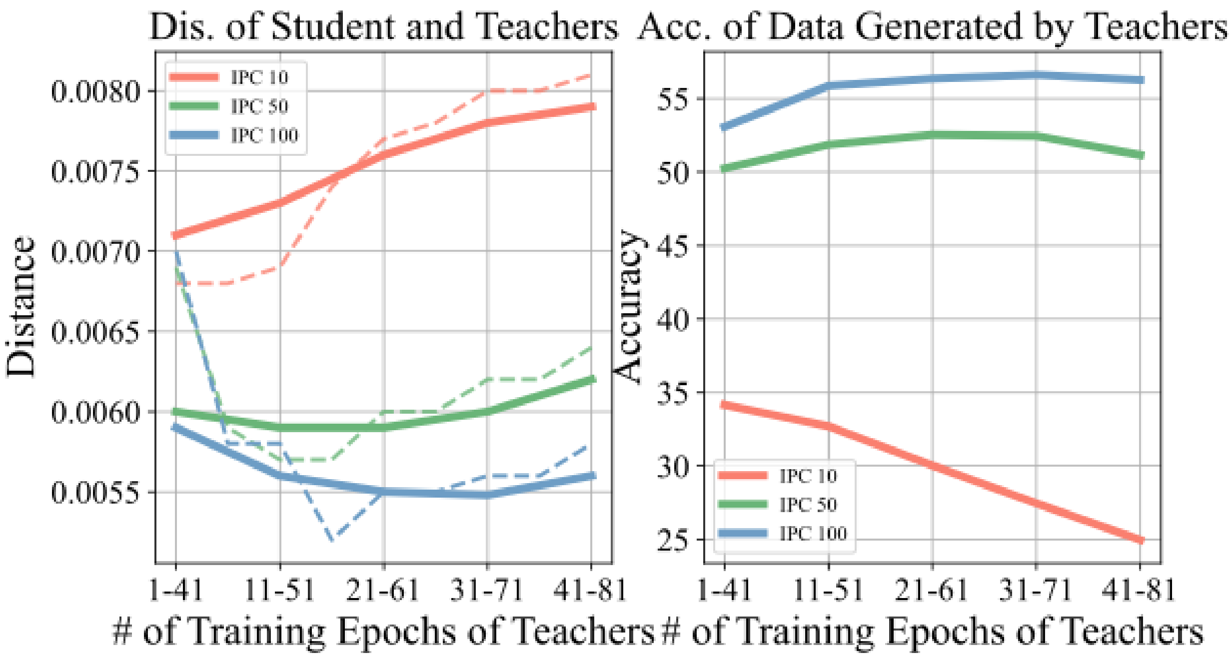
Taylor-Approximated Matching

For any segment of trajectory from $\theta_{\mathcal{S}}^{(b)}$ to $\theta_{\mathcal{S}}^{(b+e)}$, apply Taylor-approximation:

$$\mathbb{E}_{\theta^{(0)}\sim\Theta}[l_{ce}\left(\mathcal{T},\theta_{\mathcal{S}}^{(b+e)}\right)]=\mathbb{E}_{\theta^{(0)}\sim\Theta}l\left(\mathcal{T},\theta_{\mathcal{S}}^{(b)}\right)-\alpha g_{\mathcal{T}}^{(b)}(\sum_{t=b}^{b+e-1}g_{\mathcal{S}}^{(t)})$$

Single-step gradient on real dataset comparable with multi-step gradient on synthetic dataset

How to choose the teacher models?



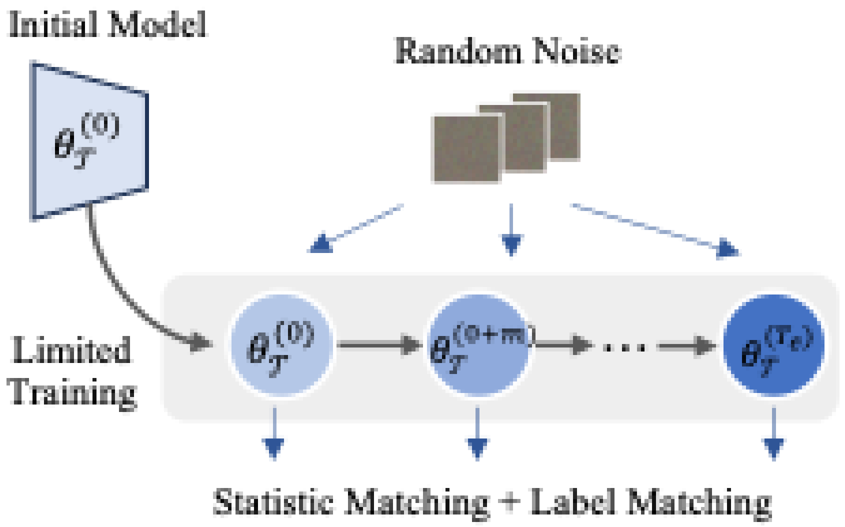
Taylor-Approximated Matching

$$\mathbb{E}_{\theta^{(0)}\sim\Theta}[l_{ce}\left(\mathcal{T},\theta_{\mathcal{S}}^{(T)}\right)] = \mathbb{E}_{\theta^{(0)}\sim\Theta}l\left(\mathcal{T},\theta_{\mathcal{S}}^{(0)}\right) - \alpha\sum_{t=0}^{T-1} g_{\mathcal{T}}^{(t)}\cdot g_{\mathcal{S}}^{(t)}$$

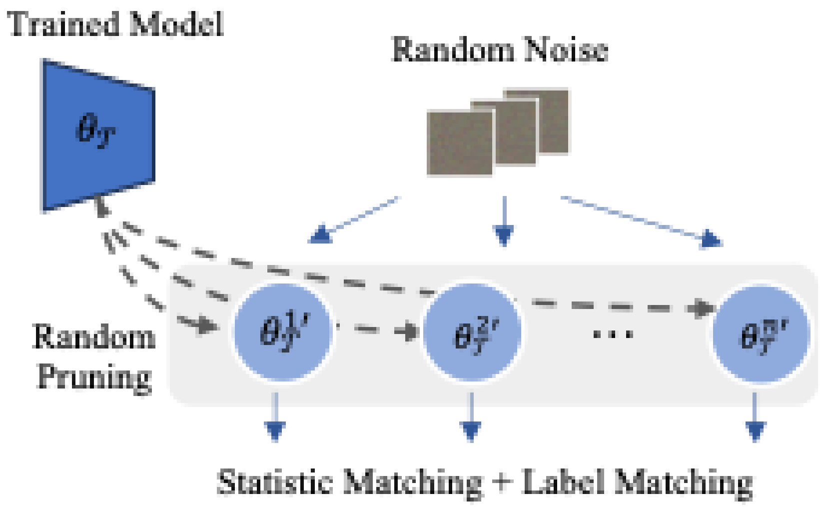
$$= \sum_{t=T_b}^{T_e}(\sum_{l}||\mu_l(f_{\theta_{\mathcal{T}}^{(t)}}(X_s) - RM_{\theta_{\mathcal{T}}^{(t)}}^{l}(X_t)|\Big|_2$$

$$+ \sum_{l}||\sigma_l^2(f_{\theta_{\mathcal{T}}^{(t)}}(X_s) - RV_{\theta_{\mathcal{T}}^{(t)}}^{l}(X_t)|\Big|_2 + u * l(\mathcal{S},\theta_{\mathcal{T}}^{(t)}))$$

Model Pool Generation



Prior Model Pool Generation



Post Model Pool Generation

Algorithm Summary

Algorithm 1: Teddy Framework

Input: Original dataset $\mathcal{T}$, single base model θ_{base}
Output: Synthetic dataset $\mathcal{S}$
Initialize $\mathcal{S}$
if θ_{base} *is from random or at early stage* **then**
  Prior-generate model pool $\mathcal{M}$
else if θ_{base} *is well-trained or at late stage* **then**
  Post-generation model pool $\mathcal{M}$
while *not converge* **do**
  Randomly select n models from $\mathcal{M}$
  Compute $\mathcal{L}(\mathcal{S}, \mathcal{T})$ as Eq. 7
  Back-propagate and update $\mathcal{S}$
Ensembly generate soft label via $\mathcal{M}$, $Y_s = \frac{1}{|\mathcal{M}|} \sum_{\theta \in \mathcal{M}} h(\mathcal{A}(X_s); \theta)$
  ▷ $h(\cdot; \theta)$ represents the model with parameter θ, $\mathcal{A}$ is the function of data augmentation.
return $\mathcal{S}$

Experimental Results

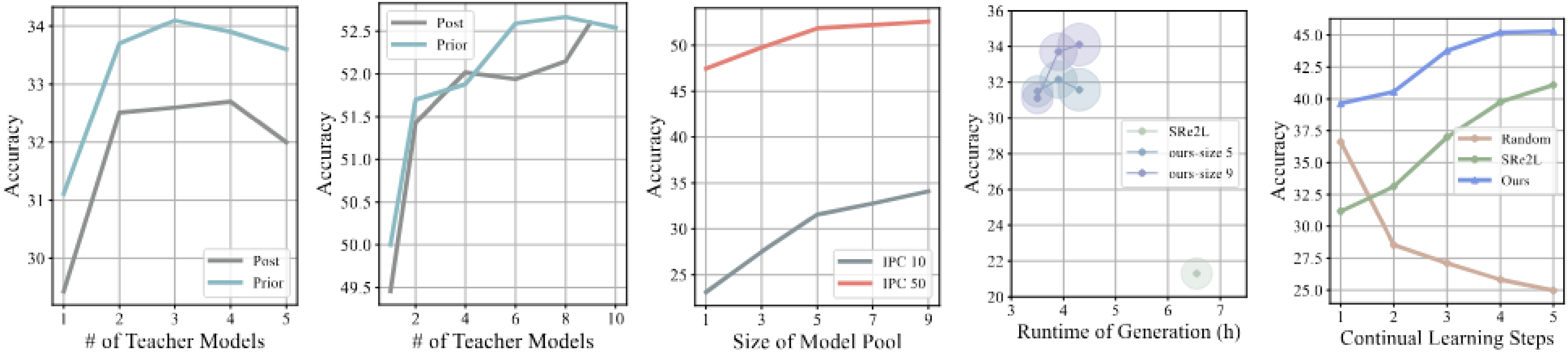
Method	Tiny-ImageNet		ImageNet-1K		
	50	100	10	50	100
Random (Conv)	15.1 ± 0.3	24.3 ± 0.3	4.1 ± 0.1*	16.2 ± 0.8*	19.5 ± 0.5*
Random (ResNet18)	18.2 ± 0.2	25.0 ± 0.2	6.8 ± 0.1	32.0 ± 0.2	45.7 ± 0.1
DC [41]	11.2 ± 0.3	-	-	-	-
DSA [39]	25.3 ± 0.2	-	-	-	-
DM [40]	24.1 ± 0.3	29.4 ± 0.2	-	-	-
IDM [42]	27.7 ± 0.3	-	-	-	-
MTT [3]	28.2 ± 0.5	33.7 ± 0.6	-	-	-
FTD [10]	31.5 ± 0.3	34.5 ± 0.4	-	-	-
TESLA [5]	33.4 ± 0.5	34.7 ± 0.2	17.8 ± 1.3*	27.9 ± 1.2*	29.2 ± 1.0*
SRe^{2}L [36]	41.1 ± 0.4	49.7 ± 0.3	21.3 ± 0.6	46.8 ± 0.2	52.8 ± 0.3
Ours (post)	44.5 ± 0.2 (+ 3.4)	51.4 ± 0.2 (+ 1.7)	32.7 ± 0.2 (+ 11.4)	**52.5 ± 0.1 (+ 5.7)**	56.2 ± 0.2 (+ 3.4)
Ours (prior)	**45.2 ± 0.1 (+ 4.1)**	**52.0 ± 0.2 (+ 2.3)**	**34.1 ± 0.1 (+ 12.8)**	**52.5 ± 0.1 (+ 5.7)**	**56.5 ± 0.1 (+ 3.7)**

Table 1: Comparison with baseline methods. * indicates the evaluation results on downsampled ImageNet-1K dataset. Here, SRe^{2}L and our proposed methods adopt the ResNet18 as the training and evaluation model, other methods adopt ConvNet.

Method	ResNet50	ResNet101	DenseNet121	MobileNetV2	ShuffleNetV2	EfficientNetB0
SRe^{2}L [36]	28.4 ± 0.1	30.9 ± 0.1	21.5 ± 0.5	10.2 ± 0.2	29.1 ± 0.1	16.1 ± 0.1
Ours (post)	37.9 ± 0.1 (+ 9.5)	40.0 ± 0.1 (+ 9.1)	33.0 ± 0.1 (+ 11.5)	20.5 ± 0.1 (+ 10.3)	**40.0 ± 0.3 (+ 10.9)**	27.3 ± 0.2 (+ 11.2)
Ours (prior)	**39.0 ± 0.1 (+ 10.6)**	**40.3 ± 0.1 (+ 9.4)**	**34.3 ± 0.1 (+ 12.8)**	**23.4 ± 0.3 (+ 13.2)**	38.5 ± 0.1 (+ 9.4)	**29.2 ± 0.1 (+ 13.1)**

Table 2: Evaluation results of cross-architecture generalization under the ImageNet-1K with IPC 10 setting. SRe^{2}L and our methods use ResNet18 as the training model.

Experimental Results



Thanks!

Contact me ruonan@u.nus.edu for further communication