NeurIPS | 2023

An Efficient Dataset Condensation Plugin and Its Application to Continual Learning

Enneng Yang[1] Li Shen[2] Zhenyi Wang[3] Tongliang Liu[4]
Guibing Guo[1]

[1]Northeastern University, China [2]JD Explore Academy, China
[3]University of Maryland, USA [4]The University of Sydney, Australia

➢ Dataset Distillation / Condensation

Definition: Dataset condensation distills a large real-world dataset into a small synthetic dataset, with the goal of training a network from scratch on the latter that *performs similarly* to the former.

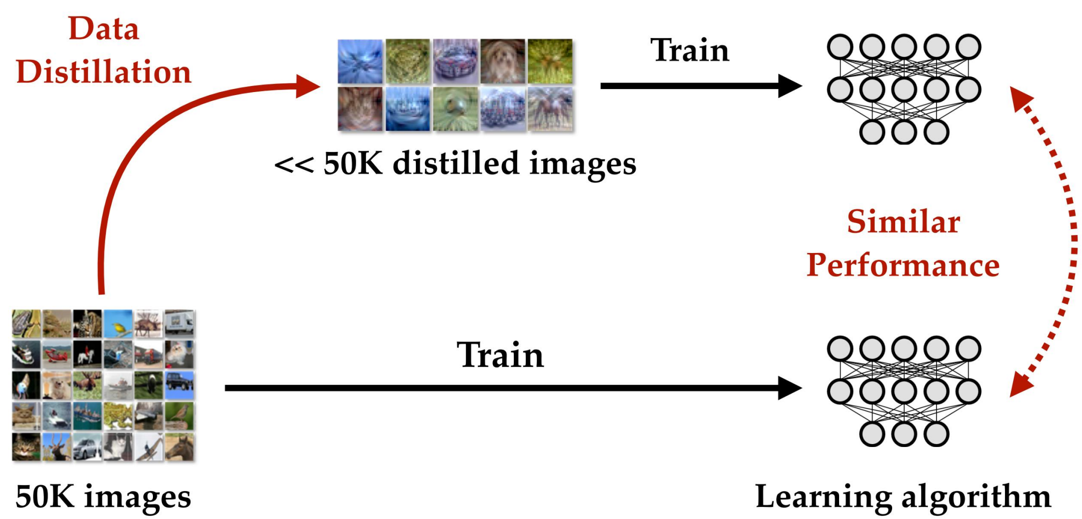


Fig.1 Dataset distillation problem paradigm[1].

[1] Data distillation: A survey. arXiv:2301.04272

NEURAL INFORMATION PROCESSING SYSTEMS
NeurIPS | 2023

➢ Problem Definition

We expect a network ϕ_{θ^S} trained on the small dataset S to have similar performance to a network ϕ_{θ^T} trained on the large training set T on the unseen test dataset, that is:

$$\mathbb{E}_{\mathbf{x}_i \sim P_\mathcal{T}}\left[\ell\left(\phi_{\theta^\mathcal{T}}(\mathbf{x}_i), y\right)\right] \simeq \mathbb{E}_{\mathbf{x}_i \sim P_\mathcal{T}}\left[\ell\left(\phi_{\theta^\mathcal{S}}(\mathbf{x}_i), y\right)\right],$$

$$\text{s.t. } \theta^\mathcal{T} = \underset{\theta^\mathcal{T}}{\arg\min}\, \mathcal{L}^\mathcal{T}(\theta^\mathcal{T}) = \underset{\theta^\mathcal{T}}{\arg\min}\, \frac{1}{N_\mathcal{T}} \sum_{(\mathbf{x}_i, y) \in \mathcal{T}} \ell\left(\phi_{\theta^\mathcal{T}}(\mathbf{x}_i), y\right),$$

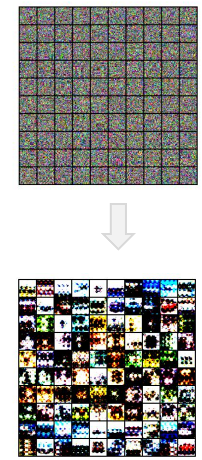
$$\theta^\mathcal{S} = \underset{\theta^\mathcal{S}}{\arg\min}\, \mathcal{L}^\mathcal{S}(\theta^\mathcal{S}) = \underset{\theta^\mathcal{S}}{\arg\min}\, \frac{1}{N_\mathcal{S}} \sum_{(\mathbf{x}_i, y) \in \mathcal{S}} \ell\left(\phi_{\theta^\mathcal{S}}(\mathbf{x}_i), y\right),$$

where P_T represents the real distribution of the test dataset.

➢ Exsting Methods

Existing DC methods [1-4] first initialize the dataset $\mathcal{S} \in \mathbb{R}^{N_S \times D \times H \times W}$ as a set of learnable parameters in high-dimensional pixel space.

- N_S: the number of synthetic images
- C: channels
- H: image's height
- W: image's width



Optimization:

In the first dataset distillation work DD [1], dataset S is treated as a hyperparameter in a bi-level optimization problem as follows:

Accuracy matching: $\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathcal{L}^{\mathcal{T}}(\phi_{\theta^{\mathcal{S}}})$, subject to $\theta^{\mathcal{S}} = \arg\min_{\theta} \mathcal{L}^{\mathcal{S}}(\phi_{\theta})$,

- **Inner loop**: Trains a randomly initialized network on the synthetic dataset S until convergence
- **Outer loop**: uses the large target dataset T as a validation set optimize S

[1] Dataset distillation. arXiv preprint arXiv:1811.10959, 2018.
[2] Dataset condensation with gradient matching. ICLR, 2021.
[3] Dataset condensation with differentiable siamese augmentation. ICML, 2021.
[4] Dataset condensation with distribution matching. WACV, 2023.

➢ Exsting Methods

SOTA DC methods are based on surrogate objectives to make the model trained on S and T approximate each other in

- **Parameter / Gradient / Distribution / ... matching**

$$\theta^{\mathcal{T}} \simeq \theta^{\mathcal{S}} \qquad \phi_\theta(\mathbf{x}_i) \simeq \phi_\theta(\mathbf{s}_i)$$

$$\nabla_\theta \mathcal{L}^{\mathcal{T}}(\theta) \simeq \nabla_\theta \mathcal{L}^{\mathcal{S}}(\theta)$$

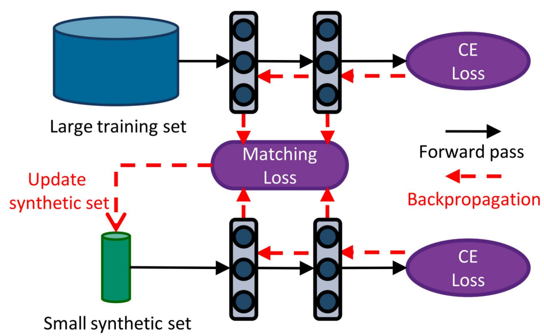


Fig. gradient matching[1].

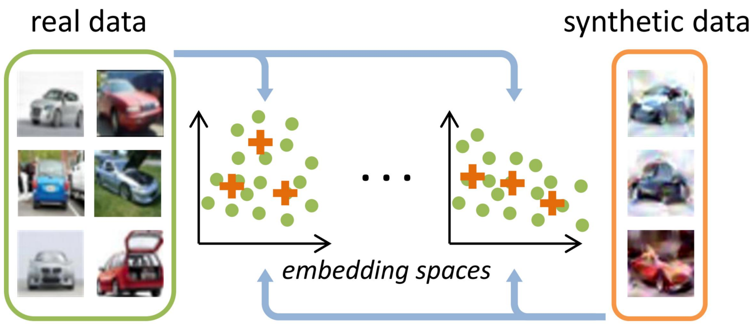
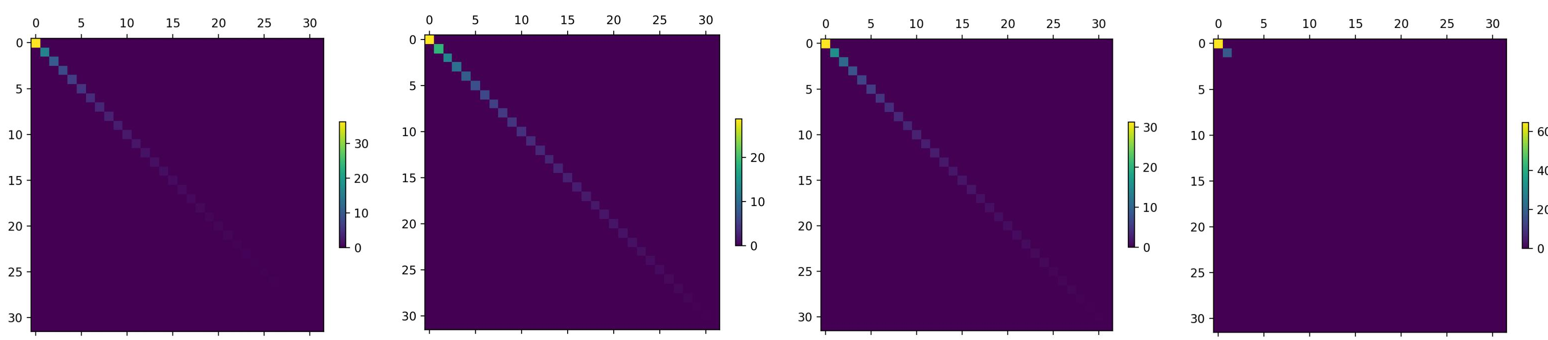


Fig. distribution matching[2].

[1] Dataset condensation with gradient matching. ICLR, 2021.
[2] Dataset condensation with distribution matching. WACV, 2023.

➢ Our Motivation



(a) Real (b) DSA (c) DM (d) LoDM

Both the real image and the image generated by traditional DC methods are low-rank, so performing DC in a high-dimensional pixel space is inefficient.

➢ Our Low-Rank Data Condensation Plugin

We conduct a low-rank decomposition of the content in each channel of an image.

Therefore, the **goal** of data condensation in the low-rank manifold is to optimize $\mathcal{A} \in \mathbb{R}^{N_S \times D \times H \times r}$ and $\mathcal{B} \in \mathbb{R}^{N_S \times D \times r \times W}$ such that the network $\phi_{\theta^{\Omega(\mathcal{A},\mathcal{B})}}$, trained on the small reconstructed data $\Omega(\mathcal{A}, \mathcal{B})$, achieves similar performance to the network $\phi_{\theta^{\mathcal{T}}}$ trained on the high-dimensional large dataset *T*.

$$\mathbb{E}_{\mathbf{x}_i \sim P_{\mathcal{T}}} \left[\ell\left(\phi_{\theta^{\mathcal{T}}}(\mathbf{x}_i), y\right)\right] \simeq \mathbb{E}_{\mathbf{x}_i \sim P_{\mathcal{T}}} \left[\ell\left(\phi_{\theta^{\Omega(\mathcal{A},\mathcal{B})}}(\mathbf{x}_i), y\right)\right],$$

$$\text{s.t. } \theta^{\mathcal{T}} = \underset{\theta^{\mathcal{T}}}{\arg\min}\, \mathcal{L}^{\mathcal{T}}(\theta^{\mathcal{T}}) = \underset{\theta^{\mathcal{T}}}{\arg\min}\, \frac{1}{N_{\mathcal{T}}} \sum_{(\mathbf{x}_i, y) \in \mathcal{T}} \ell\left(\phi_{\theta^{\mathcal{T}}}(\mathbf{x}_i), y\right),$$

$$\theta^{\Omega(\mathcal{A},\mathcal{B})} = \underset{\theta^{\Omega(\mathcal{A},\mathcal{B})}}{\arg\min}\, \mathcal{L}^{\Omega(\mathcal{A},\mathcal{B})}(\theta^{\Omega(\mathcal{A},\mathcal{B})}) = \underset{\theta^{\Omega(\mathcal{A},\mathcal{B})}}{\arg\min}\, \frac{1}{N_{\mathcal{S}}} \sum_{(\mathcal{A}_i \mathcal{B}_i, y) \in \Omega(\mathcal{A},\mathcal{B})} \ell\left(\phi_{\theta^{\Omega(\mathcal{A},\mathcal{B})}}(\mathcal{A}_i \mathcal{B}_i), y\right),$$

$$\mathbf{x}_i = \mathcal{A}_i \mathcal{B}_i = [\mathcal{A}_{i,1}\mathcal{B}_{i,1} | \ldots | \mathcal{A}_{i,D}\mathcal{B}_{i,D}] \in \mathbb{R}^{D \times H \times W}$$

➢ Incorporating Low-rank DC Plugin to SOTA Methods

Our proposed low-rank manifolds DC plugin can be easily incorporated into existing DC solutions.

- **LoDC:** Low-rank Dataset Condensation with Gradient Matching

$$\min_{\mathcal{A},\mathcal{B}} \mathrm{E}_{\theta_0 \sim P_{\theta_0}} \left[\sum_{t=1}^{T_{in}} d\left(\nabla_\theta \mathcal{L}^{\mathcal{T}}\left(\theta_t | \mathcal{T}\right), \nabla_\theta \mathcal{L}^{\Omega(\mathcal{A},\mathcal{B})}\left(\theta_t | \Omega(\mathcal{A},\mathcal{B})\right)\right)\right],$$

- **LoDM:** Low-rank Dataset Condensation with Distribution Matching

$$\min_{\mathcal{A},\mathcal{B}} \mathbb{E}_{\theta_0 \sim P_{\theta_0}} \left[d\left(\frac{1}{N_{\mathcal{T}}} \sum_{i=1}^{N_{\mathcal{T}}} \psi_{\theta_0}\left(\boldsymbol{x}_i\right), \frac{1}{N_{\mathcal{AB}}} \sum_{i=1}^{N_{\mathcal{AB}}} \psi_{\theta_0}\left(\mathcal{A}_i \mathcal{B}_i\right)\right)\right],$$

NEURAL INFORMATION PROCESSING SYSTEMS
NeurIPS | 2023

➢ Data Condensation for Deep Learning

Table 1: Comparison with coreset selection methods and dataset condensation methods.

DataSet	Img/Cls	Ratio%	Coreset Selection Methods			Dataset Condensation Methods					
			Random	Herding	Forgetting	DD	LD	DC	DSA	DM	LoDM(Ours)
MNIST	1	0.017	64.9±3.5	89.2±1.6	35.5±5.6	-	60.9±3.2	**91.7±0.5**	88.7±0.6	89.7±0.6	91.2±0.4
	10	0.17	95.1±0.9	93.7±0.3	68.1±3.3	79.5±8.1	87.3±0.7	97.4±0.2	97.1±0.1	96.5±0.2	**97.7±0.1**
	50	0.83	97.9±0.2	94.8±0.2	88.2±1.2	-	93.3±0.3	98.8±0.2	**99.2±0.1**	97.5±0.5	98.2±0.1
CIFAR10	1	0.02	14.4±2.0	21.5±1.2	13.5±1.2	-	25.7±0.7	28.3±0.5	28.8±0.7	26.0±0.8	**43.8±0.8**
	10	0.2	26.0±1.2	31.6±0.7	23.3±1.0	36.8±1.2	38.3±0.4	44.9±0.5	51.1±0.5	48.9±0.6	**59.8±0.4**
	50	1	43.4±1.0	40.4±0.6	23.3±1.1	-	42.5±0.4	53.9±0.5	60.6±0.5	63.0±0.4	**64.6±0.1**
CIFAR100	1	0.2	4.2±0.3	8.4±0.3	4.5±0.2	-	11.5±0.4	12.8±0.3	13.9±0.3	11.4±0.3	**25.6±0.5**
	10	2	14.6±0.5	17.3±0.3	15.1±0.3	-	-	25.2±0.3	32.3±0.3	29.7±0.3	**37.5±0.8**
TinyImageNet	1	0.2	1.4±0.1	2.8±0.2	1.6±0.1	-	-	4.61±0.2	4.79±0.2	3.9±0.2	**10.3±0.2**
	10	2	5.0±0.2	6.3±0.2	5.1±0.2	-	-	11.6±0.3	14.7±0.2	12.9±0.4	**18.3±0.3**

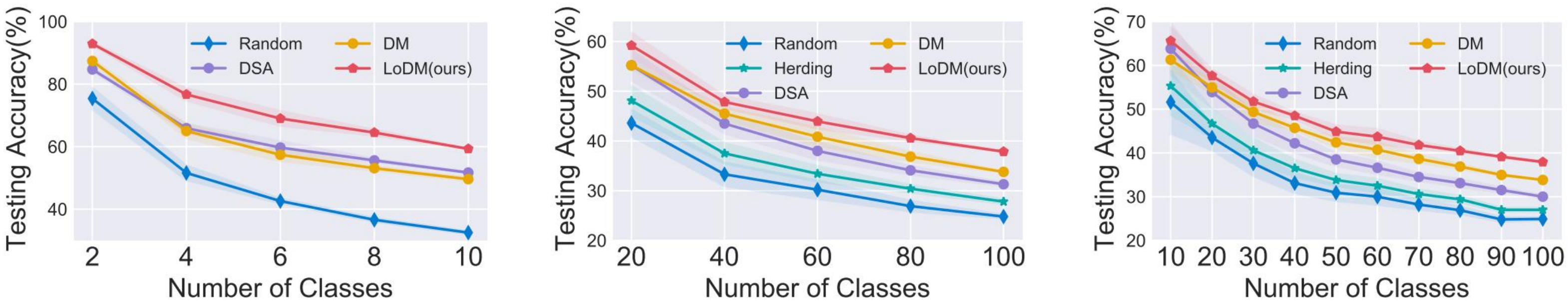
Table 4: Compare with other advanced dataset condensation methtods.

CIFAR10 (Img/Cls=1)	MTT 46.3%	IDC-I 36.7%	IDC 50.6%	HaBa 48.3%	RememberThePast 66.4%
	LoMTT 58.7%	LoIDC-I 49.2%	LoIDC 57.2%	LoHaBa 66.1%	LoRememberThePast 68.4%
CIFAR100 (Img/Cls=1)	MTT 24.3%	IDC-I 16.6%	IDC 24.9%	HaBa 33.4%	RememberThePast -
	LoMTT 31.0%	LoIDC-I 26.9%	LoIDC 33.1%	LoHaBa 36.1%	LoRememberThePast -

Observation: By utilizing the same memory, our low-rank LoDM can represent a more significant number of images, which is significantly better than other SOTA dataset compression methods, especially when the sample size of each class is small.

NEURAL INFORMATION PROCESSING SYSTEMS
NeurIPS | 2023

➢ Data Condensation for Continual Learning



(a) 5-tasks on CIFAR10 (b) 5-tasks on CIFAR100 (c) 10-tasks on CIFAR100

Observation: we observe that in the three subfigures (a-c), GDumb+LoDM achieves the best results. This suggests that our condensed data in a low-rank manifold is also meaningful for continual learning with limited memory.

Thanks！