–Supplementary Materials–
ATOM: Attention Mixer for Efficient Dataset Distillation

Samir Khaki[1]*, Ahmad Sajedi[1]*, Kai Wang[2], Lucy Z. Liu[3], Yuri A. Lawryshyn[1], and Konstantinos N. Plataniotis[1]
[1]University of Toronto [2] National University of Singapore [3]Royal Bank of Canada (RBC)
{samir.khaki, ahmad.sajedi}@mail.utoronto.ca
Code: https://github.com/DataDistillation/ATOM

1. Implementation Details

1.1. Datasets

We conducted experiments on three main datasets: CIFAR10/100 [4] and TinyImageNet [5]. These datasets are considered single-label multi-class; hence, each image has exactly one class label. The CIFAR10/100 are conventional computer vision benchmarking datasets comprising 32×32 colored natural images. They consist of 10 coarse-grained labels (CIFAR10) and 100 fine-grained labels (CIFAR100), each with 50,000 training samples and 10,000 test samples. The CIFAR10 classes include "Airplane", "Car", "Bird", "Cat", "Deer", "Dog", "Frog", "Horse", "Ship", and "Truck". The TinyImageNet dataset, a subset of ImageNet-1K [3] with 200 classes, contains 100,000 high-resolution training images and 10,000 test images resized to 64×64. The experiments on these datasets make up the benchmarking for many previous dataset distillation works [1, 2, 6, 7, 10, 11].

1.2. Dataset Pre-processing

We applied the standardized preprocessing techniques to all datasets, following the guidelines provided in DM [9] and DataDAM [6]. Following previous works, we apply the default Differentiable Siamese Augmentation (DSA) [8] scheme during distillation and evaluation. Specifically for the CIFAR10/100 datasets, we integrated Kornia zero-phase component analysis (ZCA) whitening, following the parameters outlined in [1, 6]. Similar to DataDAM [6], we opted against ZCA for TinyImagenet due to the computational bottlenecks associated with full-scale ZCA transformation on a larger dataset with double the resolution. Note that we visualized the distilled images by directly applying the inverse transformation based on the corresponding data preprocessing, without any additional modifications.

*Equal contribution

1.3. Hyperparameters

Our method conveniently introduces only one additional hyperparameter: the power term in channel attention, *i.e.* p_c. All the other hyperparameters used in our method are directly inherited from the published work, DataDAM [6]. Therefore, we include an updated hyperparameter table in Table 1 aggregating our power term with the remaining preset hyperparameters. In the main paper, we discussed the effect of power terms on both channel- and spatial-wise attention and ultimately found that higher channel attention paired with lower spatial attention works best. However, our default, as stated in the main draft, is $p_c = p_s = 4$. Regarding the distillation and train-val settings, we use the SGD optimizer with a learning rate of 1.0 for learning the synthetic images and a learning rate of 0.01 for training neural network models (for downstream evaluation). For CIFAR10/100 (low-resolution), we use a 3-layer ConvNet; meanwhile, for TinyImagenet (medium-resolution), we use a 4-layer ConvNet, following previous works in the field [1, 6, 9]. Our batch size for learning the synthetic images was set to 128 due to the computational overhead of a larger matching set.

1.4. Neural Architecture Search Details

Following previous works [6, 8–10], we define a search space consisting of 720 ConvNets on the CIFAR10 dataset. Models are evaluated on CIFAR10 using our IPC 50 distilled set as a proxy under the neural architecture search (NAS) framework. The architecture search space is constructed as a uniform grid that varies in depth $D \in \{1, 2, 3, 4\}$, width $W \in \{32, 64, 128, 256\}$, activation function $A \in \{\text{Sigmoid, ReLu, LeakyReLu}\}$, normalization technique $N \in \{\text{None, BatchNorm, LayerNorm, InstanceNorm, GroupNorm}\}$, and pooling operation $P \in \{\text{None, MaxPooling, AvgPooling}\}$ to create varying versions of the standard ConvNet. These candidate architectures are then evaluated based on their validation perfor-

mance and ranked accordingly. In the main paper, Table 6 measures various costs and performance metrics associated with each distillation method. Overall distillation improves the computational cost; however, ATOM achieves the highest correlation, which is by far the most "important" metric in this NAS search, as it indicates that our proxy set best estimates the original dataset.

2. Additional Visualizations.

We include additional visualizations of our synthetic datasets in Figure 1, Figure 2, Figure 3. The first two represent CIFAR10/100 at IPC 50, while the third depicts Tiny-ImageNet at IPC 10. Our images highly exhibit learned artifacts from the distillation process that are, in turn, helpful during downstream classification tasks.

References

[1] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022. 1

[2] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3739–3748, 2023. 1

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[4] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[5] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 1

[6] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17097–17107, 2023. 1, 6

[7] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12196–12205, 2022. 1

[8] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 1

[9] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023. 1

[10] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. 1

[11] Daquan Zhou, Kai Wang, Jianyang Gu, Xiangyu Peng, Dongze Lian, Yifan Zhang, Yang You, and Jiashi Feng. Dataset quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17205–17216, 2023. 1

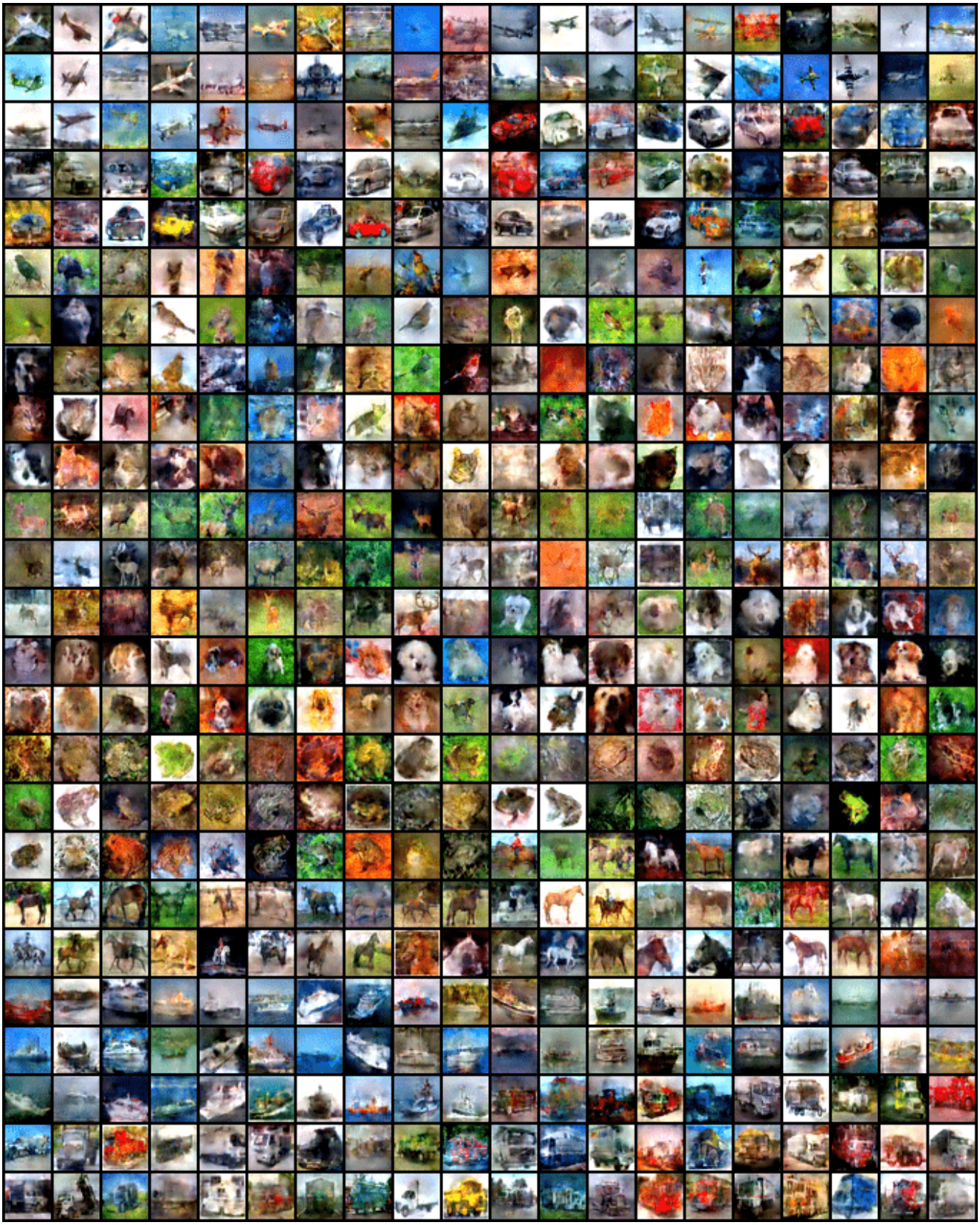

Figure 1. Distilled Image Visualization: CIFAR-10 dataset with IPC 50.

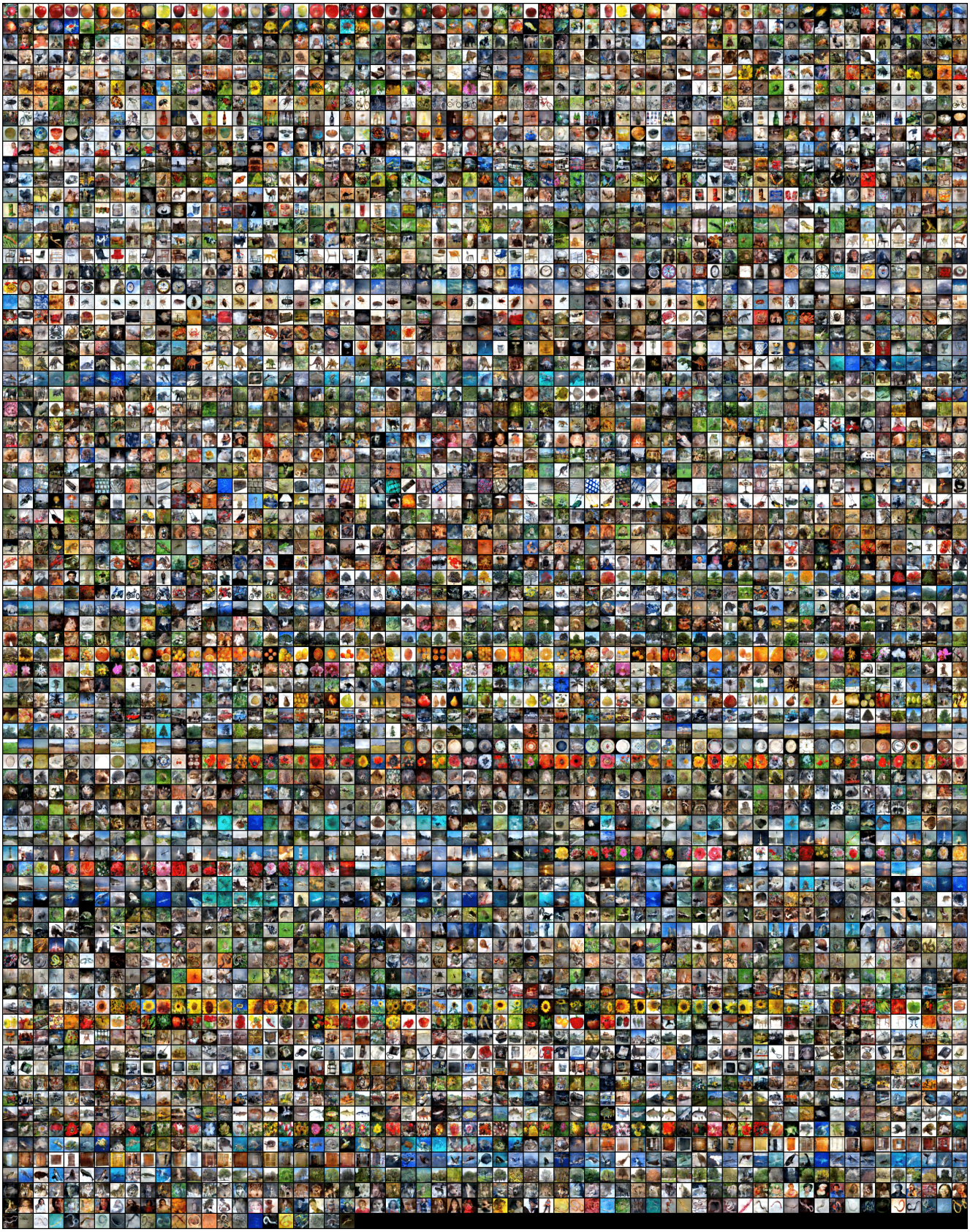

Figure 2. Distilled Image Visualization: CIFAR-100 dataset with IPC 50.

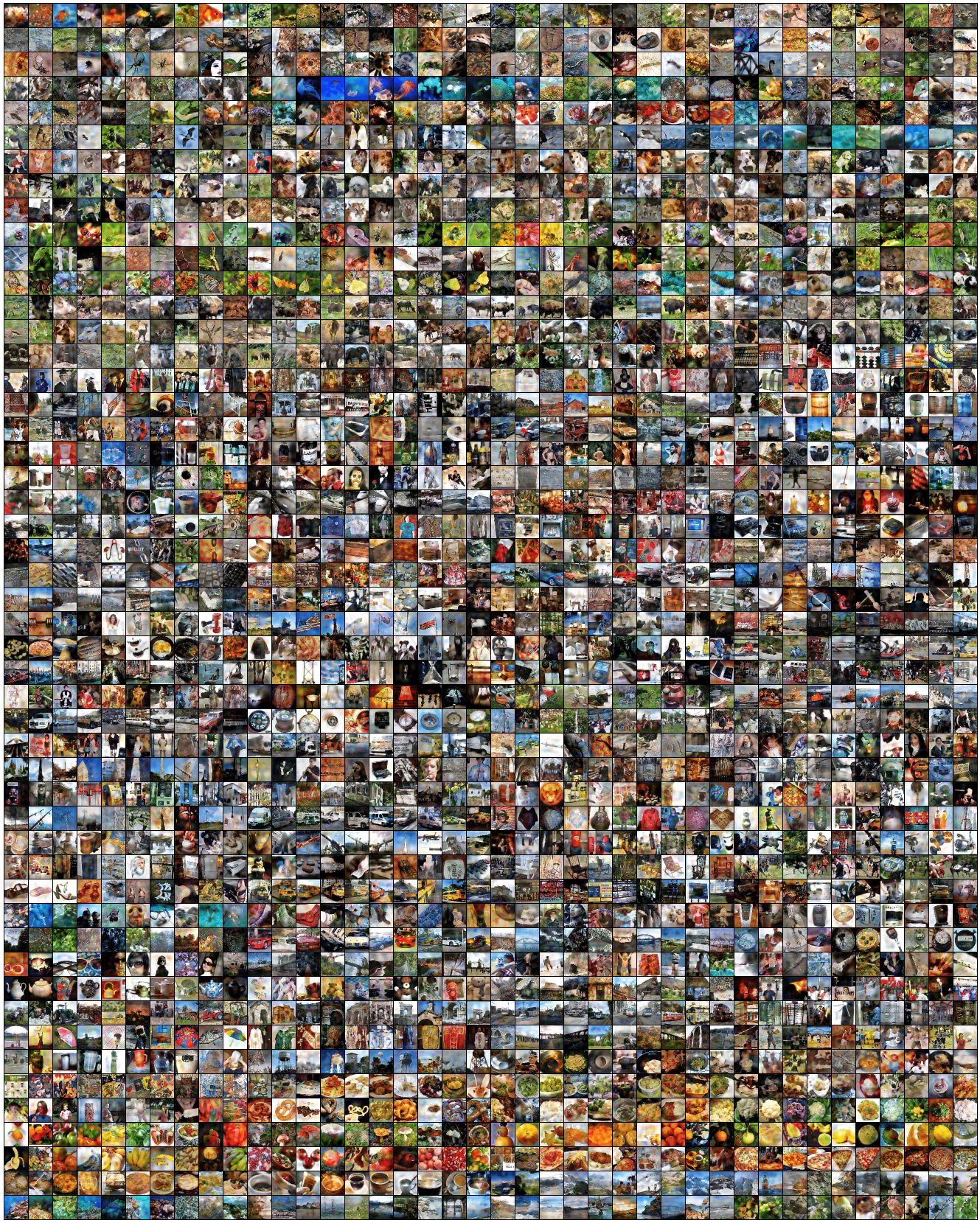

Figure 3. Distilled Image Visualization: TinyImageNet dataset with IPC 10.

Hyperparameters			Options/ Range	Value
Category	**Parameter Name**	**Description**		
Optimization	**Learning Rate $\eta_{\mathcal{S}}$ (images)**	Step size towards global/local minima	$(0, 10.0]$	IPC $\leq$ 50: 1.0 IPC $>$ 50: 10.0
	Learning Rate η_θ (network)	Step size towards global/local minima	$(0, 1.0]$	0.01
	Optimizer (images)	Updates synthetic set to approach global/local minima	SGD with Momentum	Momentum: 0.5 Weight Decay: 0.0
	Optimizer (network)	Updates model to approach global/local minima	SGD with Momentum	Momentum: 0.9 Weight Decay: $5e-4$
	Scheduler (images)	-	-	-
	Scheduler (network)	Decays the learning rate over epochs	StepLR	Decay rate: 0.5 Step size: 15.0
	Iteration Count	Number of iterations for learning synthetic data	$[1, \infty)$	8000
Loss Function	**Task Balance λ**	Regularization Multiplier	$[0, \infty)$	Low Resolution: 0.01 High Resolution: 0.02
	Spatial Power Value p_s	Exponential power for amplification of spatial attention	$[1, \infty)$	4
	Channel Power Value p_c	Exponential power for amplification of channel attention	$[1, \infty)$	4
	Loss Configuration	Type of error function used to measure distribution discrepancy	-	Mean Squared Error
	Normalization Type	Type of normalization used in the SAM module on attention maps	-	L2
DSA Augmentations	**Color**	Randomly adjust (jitter) the color components of an image	brightness saturation contrast	1.0 2.0 0.5
	Crop	Crops an image with padding	ratio crop pad	0.125
	Cutout	Randomly covers input with a square	cutout ratio	0.5
	Flip	Flips an image with probability p in range:	$(0, 1.0]$	0.5
	Scale	Shifts pixels either column-wise or row-wise	scaling ratio	1.2
	Rotate	Rotates image by certain angle	$0^\circ - 360^\circ$	$[-15^\circ, +15^\circ]$
Encoder Parameters	**Conv Layer Weights**	The weights of convolutional layers	$\mathbb{R}$ bounded by kernel size	Uniform Distribution
	Activation Function	The non-linear function at the end of each layer	-	ReLU
	Normalization Layer	Type of normalization layer used after convolutional blocks	-	InstanceNorm

Table 1. Hyperparameters Details – boilerplate obtained from DataDAM [6].