A. Hyperparameter Settings

Table 7. G-VBSM hyperparameter settings on ImageNet-1k.

Phase	Optimizer	Learning Rate	Optimizer Momentum	Loss Function	Batch Size	Epoch/ Iteration	Augmentation	Others
Pre-trained model training	SGD	0.1	0.9	cross-entropy	256	Epoch 100	RandomResizedCrop	-
Data synthesis	Adam	0.1	β_1,β_2=0.5,0.9	$\ell(f_{\text{cand}}(\tilde{X}), y) + \mathcal{L}'_{\text{BN}} + \mathcal{L}_{\text{DD}} + \mathcal{L}'_{\text{Conv}}$	40	Iteration 4000	RandomResizedCrop	β_{dr}=0.4, Backbone= {ResNet18,MobileNetV2 ,EfficientNet-B0,ShuffleNetV2-0.5}
Soft label generation	-	-	-	-	1024	Epoch 300	RandomResizedCrop, CutMix	Backbone= {ResNet18,MobileNetV2 ,EfficientNet-B0,ShuffleNetV2-0.5}
Evaluation	AdamW	0.001	β_1,β_2=0.9,0.999	MSE+0.1×GT	1024	Epoch 300	RandomResizedCrop, CutMix	Evaluation Model= {ResNet18,ResNet50,ResNet101, MobileNetV2,Swin-Tiny,DeiT-Tiny}

Table 8. G-VBSM hyperparameter settings on Tiny-ImageNet.

Phase	Optimizer	Learning Rate	Optimizer Momentum	Loss Function	Batch Size	Epoch/ Iteration	Augmentation	Others
Pre-trained model training	SGD	0.1	0.9	cross-entropy	128	Epoch 50	RandomCrop RandomHorizontalFlip	-
Data synthesis	Adam	0.05	β_1,β_2=0.5,0.9	$\ell(f_{\text{cand}}(\tilde{X}), y) + \mathcal{L}'_{\text{BN}} + \mathcal{L}_{\text{DD}} + \mathcal{L}'_{\text{Conv}}$	50	Iteration 4000	RandomResizedCrop	β_{dr}=0.4, Backbone= {ResNet18,128-width ConvNet,MobileNetV2 ,WRN-16-2,ShuffleNetV2-0.5}
Soft label generation	-	-	-	-	128	Epoch 100	RandomResizedCrop, CutMix	Backbone= {ResNet18,128-width ConvNet,MobileNetV2 ,WRN-16-2,ShuffleNetV2-0.5}
Evaluation	SGD	0.2, 0.1 and 0.1 on ResNet18, ResNet50 and ResNet101 , respectively	0.9	MSE+0.1×GT	128	Epoch 100	RandomResizedCrop, CutMix	Evaluation Model= {ResNet18,ResNet50,ResNet101}

Table 9. G-VBSM hyperparameter settings on CIFAR-10/100.

Phase	Optimizer	Learning Rate	Optimizer Momentum	Loss Function	Batch Size	Epoch/ Iteration	Augmentation	Others
Pre-trained model training	SGD	0.05	0.9	cross-entropy	64	Epoch 50 and 5 on CIFAR-100 and CIFAR-10 , respectively	RandomCrop RandomHorizontalFlip	-
Data synthesis	Adam	0.05	β_1,β_2=0.5,0.9	$\ell(f_{\text{cand}}(\tilde{X}), y) + \mathcal{L}'_{\text{BN}} + \mathcal{L}_{\text{DD}} + \mathcal{L}'_{\text{Conv}}$	50	Iteration 4000	RandomResizedCrop	β_{dr}=0.4, Backbone= {ResNet18,128-width ConvNet,MobileNetV2 ,WRN-16-2,ShuffleNetV2-0.5}
Soft label generation	-	-	-	-	64 or $\lvert\mathcal{S}\rvert$ ($\lvert\mathcal{S}\rvert \leq$100)	Epoch 1000	RandomResizedCrop, CutMix	Backbone= {ResNet18,128-width ConvNet,MobileNetV2 ,WRN-16-2,ShuffleNetV2-0.5}
Evaluation	SGD and AdamW on CIFAR-100 and CIFAR-10 , respectively	0.1 and 0.001 on CIFAR-100 and CIFAR-10 , respectively	0.9 and β_1,β_2=0.9,0.999 on CIFAR-100 and CIFAR-10 , respectively	MSE+0.15×GT	64 or $\lvert\mathcal{S}\rvert$ ($\lvert\mathcal{S}\rvert \leq$100)	Epoch 1000	RandomResizedCrop, CutMix	Evaluation Model= {128-width ConvNet,ResNet18}

Here, we present the hyperparameter settings of G-VBSM in the pre-trained model training (*i.e.*, Squeeze in SRe2L), the data synthesis (*i.e.*, Recover in SRe2L), the soft label generation (*i.e.*, Relabel in SRe2L), and the evaluation phases in Tables 7 (ImageNet-1k), 8 (Tiny-ImageNet), and 9 (CIFAR-10/CIFAR-100). The hyperparameter settings for the ImageNet-1k and Tiny-ImageNet datasets predominantly adhere to SRe2L [34]. Furthermore, the settings for CIFAR-10/CIFAR-100 draw upon the classical knowledge distillation framework [6, 8, 27, 43]. Notably, we employ the same evaluation model (*i.e.* 128-width ConvNet) and identical number of epochs (*i.e.* 1000) during the evaluation phase on CIFAR-10/CIFAR-100 as those used in the prior dataset distillation approaches [1, 20], ensuring experimental fairness.

The Consistency of Backbone used in Data Synthesis and Soft Label Generation. In all experiments conducted on different datasets, we maintain the same architectures and identical parameters of the pre-trained model for data synthesis and soft label generation. Similar to SRe2L, our exploratory studies revealed that preserving the consistency of the backbone results in the best generalization ability for the distilled dataset.

The Hyperparameter β_{dr}. Given G-VBSM's computational efficiency on ImageNet-1k under IPC 10, which serves as the benchmark for the majority of our ablation studies, we set β_{dr} to 0.0 for this specific benchmark. For the remaining experiments, including Tiny-ImageNet, CIFAR-10 and CIFAR-100, β_{dr} is set to 0.4.

The Weights of the Loss Function. To underscore the generalization and applicability of our proposed G-VBSM, we intentionally avoid setting the weights of any loss functions, except for MSE+$\gamma\times$GT, in a bespoke manner. To be specific, the weights for both $\mathcal{L}'_{\text{BN}}$ and $\mathcal{L}'_{\text{Conv}}$ are established at 0.01 for ImageNet-1k, consistent with the weight of $\mathcal{L}_{\text{BN}}$, which SRe2L is set as 0.01 for ImageNet-1k. Since we transposed the loop (*i.e.*, translate the original loop to the reorder loop), the weights for $\mathcal{L}'_{\text{BN}}$ and $\mathcal{L}'_{\text{Conv}}$ are set at 0.01 for Tiny-ImageNet, different from SRe2L, which assigns a weight of 1.0 to $\mathcal{L}_{\text{BN}}$ for the same dataset. Due to SRe2L was not evaluated on CIFAR-10 and CIFAR-100, we empirically adjusted the weights for $\mathcal{L}'_{\text{BN}}$ and $\mathcal{L}'_{\text{Conv}}$ to 0.01 in our experiments. Additionally, the weight of $\mathcal{L}_{\text{DD}}$ is consistently applied at 1.0 across all datasets. Through empirical validation in our experiments, we establish that the performance of the distilled dataset – when the weight of $\mathcal{L}_{\text{DD}}$ is configured as $\{0.1, 1.0, 10.0\}$ – remains precisely identical.

B. The Derivation of "match in the form of score distillation sampling"

Our proposed novel loss function, denoted as $\mathcal{L}'_{\text{BN}}(\tilde{X})$, draws inspiration from score distillation sampling (**SDS**) [17]. It is employed to mitigate the performance degradation that arises from directly substituting the original loop with the reorder loop. Here, we give the detailed derivation of $\mathcal{L}'_{\text{BN}}(\tilde{X})$ to facilitate understanding. First, we give the gradient of the original loss function $\mathcal{L}_{\text{BN}}(\tilde{X})$ with respect to $\tilde{X}$:

$$\frac{\partial \mathcal{L}_{\text{BN}}(\tilde{X})}{\partial \tilde{X}} = \sum_l \frac{\partial \mu_l(\tilde{X})}{\partial \tilde{X}} \frac{\mu_l(\tilde{X}) - \mathbf{BN}_l^{\text{CM}}}{||\mu_l(\tilde{X}) - \mathbf{BN}_l^{\text{CM}}||_2} + \frac{\partial \sigma_l^2(\tilde{X})}{\partial \tilde{X}} \frac{\sigma_l^2(\tilde{X}) - \mathbf{BN}_l^{\text{CV}}}{||\sigma_l^2(\tilde{X}) - \mathbf{BN}_l^{\text{CV}}||_2}. \tag{12}$$

In Eq. 12, $\frac{\mu_l(\tilde{X})-\mathbf{BN}_l^{\text{CM}}}{||\mu_l(\tilde{X})-\mathbf{BN}_l^{\text{CM}}||_2}$ and $\frac{\sigma_l^2(\tilde{X})-\mathbf{BN}_l^{\text{CV}}}{||\sigma_l^2(\tilde{X})-\mathbf{BN}_l^{\text{CV}}||_2}$ are unit vectors that dominate the direction of the gradient descent in the data synthesis process. As analyzed in Sec.3.1, the precise global statistics generated by all past batches are feasible to assist in matching between the limited statistics generated by the current batch and $\mathbf{BN}_l^{\text{CM}}$ as well as $\mathbf{BN}_l^{\text{CV}}$. We utilize EMA to update the statistics μ_l^{total} and $\sigma_l^{2,\text{total}}$ generated by all past batches:

$$\mu_l^{\text{total}} = \alpha\mu_l^{\text{total}} + (1-\alpha)\mu_l(\tilde{X}), \sigma_l^{2,\text{total}} = \alpha\sigma_l^{2,\text{total}} + (1-\alpha)\sigma_l^2(\tilde{X}). \tag{13}$$

We can achieve the SDS-like loss by simply replacing $\frac{\mu_l(\tilde{X})-\mathbf{BN}_l^{\text{CM}}}{||\mu_l(\tilde{X})-\mathbf{BN}_l^{\text{CM}}||_2}$ with $\frac{\mu_l^{\text{total}}(\tilde{X})-\mathbf{BN}_l^{\text{CM}}}{||\mu_l^{\text{total}}(\tilde{X})-\mathbf{BN}_l^{\text{CM}}||_2}$ and $\frac{\sigma_l^2(\tilde{X})-\mathbf{BN}_l^{\text{CV}}}{||\sigma_l^2(\tilde{X})-\mathbf{BN}_l^{\text{CV}}||_2}$ with $\frac{\sigma_l^{2,\text{total}}(\tilde{X})-\mathbf{BN}_l^{\text{CV}}}{||\sigma_l^{2,\text{total}}(\tilde{X})-\mathbf{BN}_l^{\text{CV}}||_2}$. In this way, the direction of gradient descent for data synthesis is no longer determined by the imprecise statistics of the single current batch, which ultimately improves the quality of the synthetic data and its ability to generalize to unseen evaluation models. In practice, we can implement the replacement easily with Pytorch's [16] stop_grad$(\cdot)$ operator:

$$\mathcal{L}'_{\text{BN}}(\tilde{X}) = \sum_l ||\mu_l(\tilde{X}) - \mathbf{BN}_l^{\text{CM}} - \text{stop\_grad}(\mu_l(\tilde{X}) - \mu_l^{\text{total}})||_2 + ||\sigma_l^2(\tilde{X}) - \mathbf{BN}_l^{\text{CV}} - \text{stop\_grad}(\sigma_l^2(\tilde{X}) - \sigma_l^{2,\text{total}})||_2. \tag{14}$$

We can find the gradient of $\mathcal{L}'_{\text{BN}}(\tilde{X})$ with respect to $\tilde{X}$ by derivation as

$$\frac{\partial \mathcal{L}'_{\text{BN}}(\tilde{X})}{\partial \tilde{X}} = \sum_l \frac{\partial \mu_l(\tilde{X})}{\partial \tilde{X}} \frac{\mu_l^{\text{total}}(\tilde{X}) - \mathbf{BN}_l^{\text{CM}}}{||\mu_l^{\text{total}}(\tilde{X}) - \mathbf{BN}_l^{\text{CM}}||_2} + \frac{\partial \sigma_l^2(\tilde{X})}{\partial \tilde{X}} \frac{\sigma_l^{2,\text{total}}(\tilde{X}) - \mathbf{BN}_l^{\text{CV}}}{||\sigma_l^{2,\text{total}}(\tilde{X}) - \mathbf{BN}_l^{\text{CV}}||_2}. \tag{15}$$

Clearly, $\mathcal{L}'_{\text{BN}}(\tilde{X})$ effectively achieves our primary purpose. Additionally, our ablation studies in Sec. 4.1 empirically demonstrate that the "match in the form of score distillation sampling" strategy is remarkable.

C. Additional Ablation Experiments on ImageNet-1k

$\mathcal{L}'_{\text{BN}}$	$\mathcal{L}'_{\text{Conv}}$	ResNet18	ResNet50	ResNet101
✓		27.8%	33.4%	35.5%
	✓	24.0%	26.1%	30.4%
✓	✓	31.4%	35.4%	38.2%

Table 10. Ablation study about $\mathcal{L}'_{\text{BN}}$ and $\mathcal{L}'_{\text{Conv}}$ in the synthetic data phase on ImageNet-1k. Meanwhile, ResNet $\{18, 50, 101\}$ represent evaluation models.

This section presents ablation experiments for $\mathcal{L}'_{\text{BN}}$ and $\mathcal{L}'_{\text{Conv}}$ to underscore their equal importance. As illustrated in Table 10, omitting either $\mathcal{L}'_{\text{BN}}$ or $\mathcal{L}'_{\text{Conv}}$ from the entire loss function during data synthesis phase leads to a performance decline. Hence, conducting the "local-match-global" matching via both $\mathcal{L}'_{\text{BN}}$ and $\mathcal{L}'_{\text{Conv}}$ is essential.

D. Exploratory Studies on CIFAR-10/100

The Choice of Candidate Backbones in GBM. Under IPC 10 on CIFAR-100, we evaluated candidate backbones {ResNet18, MobileNetV2, WRN-16-2, ShuffleNetV2-0.5}, omitting the 128-width ConvNet, during the data synthesis and soft label generation phases. In addition, we kept other hyperparameters consistent as shown in Table 9 and obtained the 128-width ConvNet evaluation performance as 32.8%. However, incorporating the 128-width ConvNet into the candidate backbones increased the accuracy from 32.8% to 38.7%. It's important to mention that the 128-width ConvNet solely utilizes GroupNorm, not BatchNorm. This enhancement to 38.7% was accomplished by relying solely on statistics within Convolution, substantiating that statistics in BatchNorm may not be the only option in the data synthesis phase.

Evaluation Model\Epoch	5	10	20	40
128-width ConvNet	46.5%	45.8%	42.5%	42.1%

Table 11. Ablation study about the number of epochs in pre-trained model training phase. We maintain the consistency of other hyperparameters as presented in Table 9.

The Number of Epochs in the Pre-Trained Model Training Phase. As illustrated in Table 11, fewer pre-training epochs on CIFAR-10 enhance the generalization of the distilled dataset. This finding could provide an explanation for the remarkable performance achieved by traditional algorithms [1, 32] on CIFAR-10, even when they employ models with random initializations. As a result, this ablation study informed our decision to pre-train models on CIFAR-10 for only 5 epochs. More important, as our experiments transition from ImageNet-1k to Tiny-ImageNet to CIFAR-100, and finally to CIFAR-10, the dataset complexity reduces, and the ideal number of pre-training epochs successively decreases from 100 to 50, to 50, and finally to 5. The most intuitive and empirical extrapolation is due to the complexity of the dataset, and we believe that this conclusion may be of some inspiration to other researchers.

E. Additional Explanation of Data Densification

Here we provide theoretical proofs within Eq. 16 to show that entropy $H(\Sigma_y/\tau)$ $(\tau > 1)$ is greater than $H(\Sigma_y)$, thus increasing $H(\Sigma_y)$ through Eq. 5, which ultimately improves the entropy of the eigenvalues and ensures the diversity of data.

$$
\begin{aligned}
&H(z/\tau) - H(z) = (\tau - 1)\frac{\partial H}{\partial \tau}(\tau'),\ s.t.\ 1 \le \tau' \le \tau, \text{define } z = \Sigma_y \text{for convenience} \\
&= -\frac{\tau - 1}{\tau'} \sum_i [(\log(\frac{e^{z_i/\tau'}}{\sum_j e^{z_j/\tau'}}) + 1)(\frac{e^{z_i/\tau'}}{\sum_j e^{z_j/\tau'}}) \\
&(\frac{-z_i/\tau' \sum_j (e^{z_j/\tau'}) + \sum_j (e^{z_j/\tau'} z_j/\tau')}{\sum_j e^{z_j/\tau'}})] = -\frac{\tau - 1}{\tau'} [\sum_i \log(\frac{e^{z_i/\tau'}}{\sum_j e^{z_j/\tau'}}) \\
&(\frac{e^{z_i/\tau'}}{\sum_j e^{z_j/\tau'}})(\frac{-z_i/\tau' \sum_j (e^{z_j/\tau'}) + \sum_j (e^{z_j/\tau'} z_j/\tau')}{\sum_j e^{z_j/\tau'}}) - \cancel{\frac{\sum_j z_j/\tau' e^{z_j/\tau'}}{\sum_j e^{z_j/\tau'}}} \\
&+ \cancel{\frac{\sum_j z_j/\tau' e^{z_j/\tau'}}{\sum_j e^{z_j/\tau'}}}] = -\frac{\tau - 1}{\tau'} \sum_j (\frac{e^{z_i/\tau'}}{\sum_j e^{z_j/\tau'}}) \log(\frac{e^{z_i/\tau'}}{\sum_j e^{z_j/\tau'}}) \\
&[-z_i/\tau' + \frac{\sum_j e^{z_j/\tau'} z_j/\tau'}{\sum_j e^{z_j/\tau'}}] = \frac{\tau - 1}{\tau'} [\sum_i (z_i/\tau')^2 e^{z_i/\tau'} - \sum_i (z_i/\tau' e^{z_i/\tau'}) \\
&\frac{\sum_i (z_i/\tau' e^{z_i/\tau'})}{\sum_i (e^{z_i/\tau'})} + \log(\sum_i e^{z_i/\tau'})(\cancel{\sum_i z_i/\tau' e^{z_i/\tau'}} - \cancel{\sum_i z_i/\tau' e^{z_i/\tau'}})] > 0.
\end{aligned}
\tag{16}
$$

F. Statistics Visualization

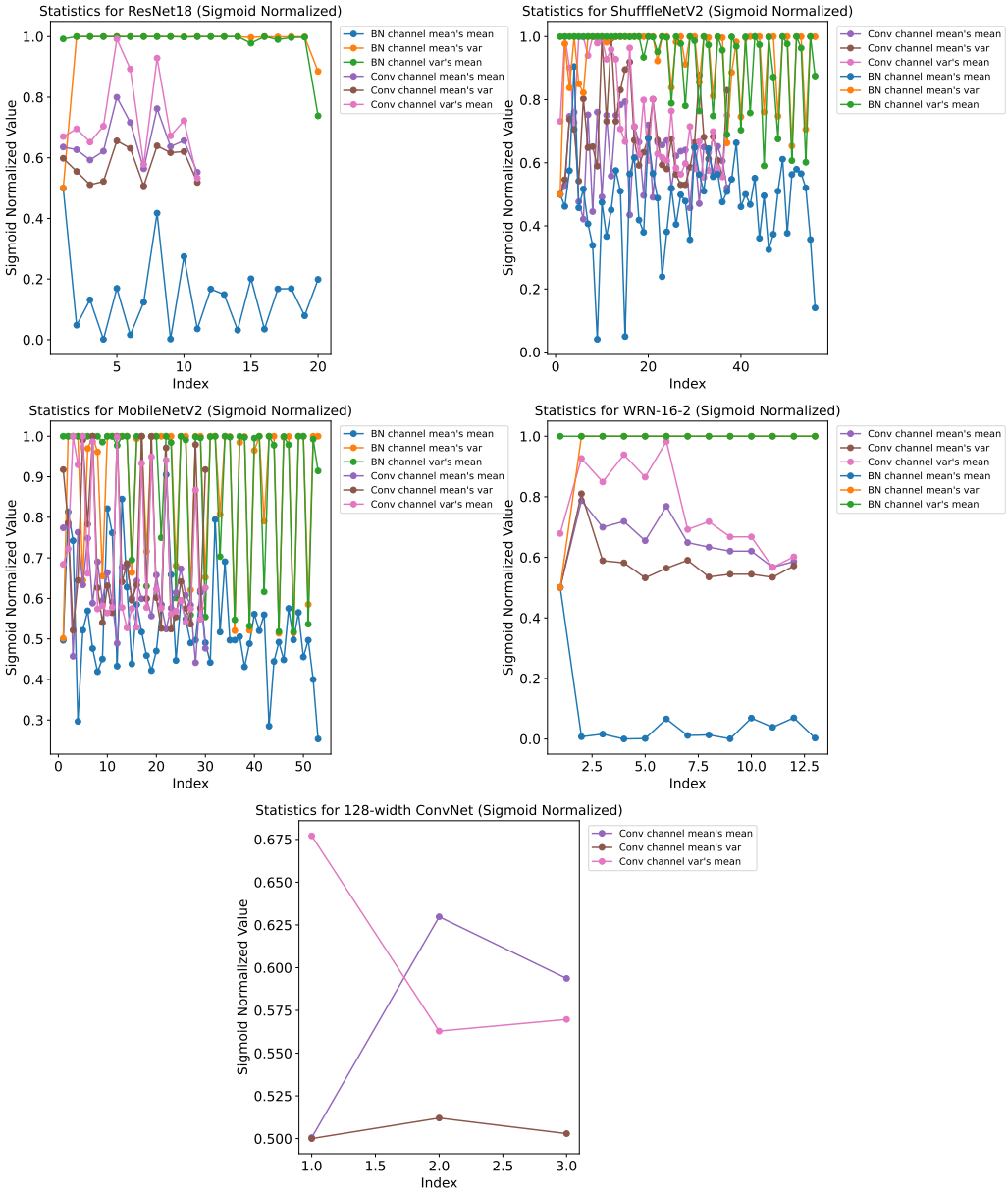

Figure 9. Statistics visualization across various backbones {ResNet18, ShuffleNetV2, MobileNetV2, WRN-16-2, 128-width ConvNet} on CIFAR-100.

It is important to emphasize that Convolution and BatchNorm offer different supervisory information in statistics. As a result, G-VBSM is more effective than SRe2L when optimized statistics in Convolution and BatchNorm together. For clarity, we visualized the statistics in the pre-trained models {ResNet18, 128-width ConvNet, MobileNetV2, WRN-16-2, ShuffleNetV2-0.5} on CIFAR-100 in Fig. 9. In each subplot of Fig. 9, the horizontal axis denotes the layer index (with orthogonal indexes for Convolution and BatchNorm), while the vertical axis shows the post-sigmoid normalized result. Due to the extensive dimensions of channel mean and channel variance, we calculate only their mean and variance for visualization. Furthermore, since BatchNorm is not included in 128-width ConvNet, only Convolution statistics are visualized. From Fig. 9, we can

conclude that the values of the statistics in Convolution and BatchNorm are different in any model, which indicates that G-VBSM is significant and can enhance the generalization of the distilled dataset as demonstrated in Fig. 5.

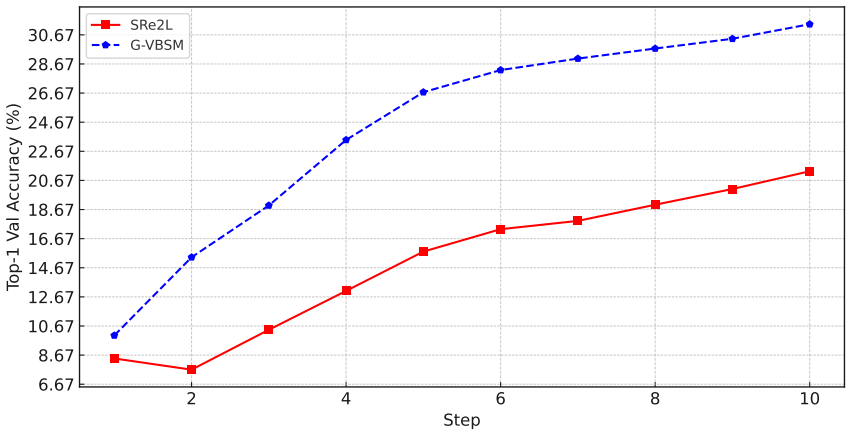

Figure 10. The experimental result of continual learning application on ImageNet-1k under IPC 10.

G. Continual Learning Application

Many data condensation algorithms [34, 38, 41] have evaluated the generalization ability of distilled datasets in continual learning. We follow the class-incremental learning approach[2] adopted in DM [41] for performing this task. Similar to the ablation studies of the main paper, our experiments are conducted on the full 224×224 ImageNet-1k, underscoring that G-VBSM is intended for use with large-scale datasets. We conduct class incremental learning with ResNet18 on the 10-step class-incremental learning strategy under 10 IPC. The experimental results are illustrated in Fig. 10. We can discover that G-VBSM significantly outperforms SRe2L, thus confirming the usefulness and effectiveness of G-VBSM.

H. Date Free Pruning Application

Data Free Pruning (ImageNet-1k, VGG-A, 50% Pruned)	IPC 10 SRe2L	IPC 10 SRe2L+DD	IPC 50 SRe2L	IPC 50 SRe2L+DD
Top-1 Val Accuracy	12.5%	12.9%	31.7%	32.8%

Table 12. The experimental result of data free pruning application on IamgeNet-1k.

Data Free Pruning of Slimming aims to reduce the model size and decrease the run-time memory footprint simultaneously for convolutional nerual network. We argue that the distilled dataset facilitates efficient data-free pruning. To substantiate this claim, we conduct experiments on ImageNet-1k with IPC 10. As illustrated in Table 12, data densification enhances downstream knowledge transfer as above by increasing synthesized data diversity and significantly boosting SRe2L.

I. Synthetic Data Visualization

We provide more visualization results on synthetic data randomly selected from G-VBSM in Figs. 11 (ImageNet-1k), 12 (Tiny-ImageNet), 13 (CIFAR-100) and 14 (CIFAR-10).

[2]This involves gradually increasing the number of classes and combining previously stored data with newly acquired data to train a model from scratch.

Figure 11. Synthetic data visualization on ImageNet-1k randomly selected from G-VBSM.

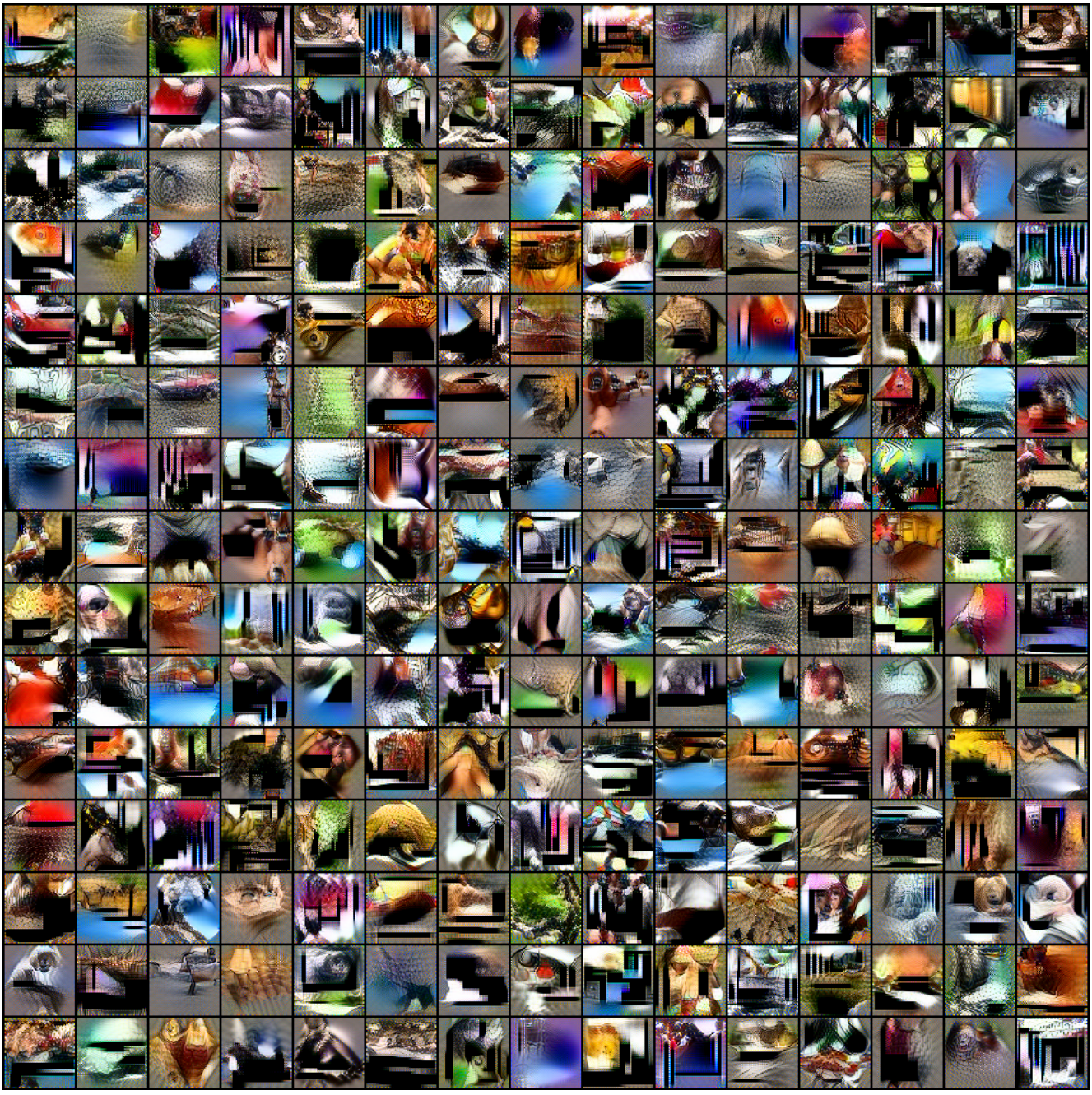

Figure 12. Synthetic data visualization on Tiny-ImageNet randomly selected from G-VBSM.

Figure 13. Synthetic data visualization on CIFAR-100 randomly selected from G-VBSM.

Figure 14. Synthetic data visualization on CIFAR-10 randomly selected from G-VBSM.