Squeeze, *Recover* and *Relabel*: Dataset Condensation at ImageNet Scale From A New Perspective Supplementary Material

Anonymous Author(s)
Affiliation
Address
email

Appendix

In the appendix, we provide details omitted in the main text, including:

- Section A: Implementation Details.
- Section B: Low-Resolution Data (32×32).
- Section C: Feature Embedding Distribution.
- Section D: More Visualization of Synthetic Data.

A Implementation Details

A.1 Dataset Statistics

Table 1 enumerates various permutations of ImageNet-1K training set, delineated according to their individual configurations. Tiny-ImageNet [1] incorporates 200 classes derived from ImageNet-1K, with each class comprising 500 images possessing a resolution of 64×64. ImageNette/ImageWoof [2] (alternatively referred to as subsets of ImageNet) include 10 classes from analogous subcategories, with each image having a resolution of 112×112. The MTT [3] framework introduces additional 10-class subsets of ImageNet, encompassing ImageFruit, ImageSquawk, ImageMeow, ImageBlub, and ImageYellow. ImageNet-10/100 [4] samples 10/100 classes from ImageNet while maintaining an image resolution of 224×224. Downsampled ImageNet-1K rescales the entirety of ImageNet data to a resolution of 64×64. In our experiments, we opt for two standard datasets of relatively large scale: Tiny-ImageNet and the full ImageNet-1K.

Training Dataset	#Class	#Img per class	Resolution	Method
Tiny-ImageNet [1]	200	500	64×64	MTT [3], FRePo [5], DM [6], SRe^{2}L (Ours)
ImageNette/ImageWoof [2]	10	∼1,000	112×112	MTT [3], FRePo [5]
ImageNet-10/100 [4]	10/100	∼1,200	224×224	IDC [7]
Downsampled ImageNet-1K [8]	1,000	∼1,200	64×64	TESLA [9], DM [6]
Full ImageNet-1K [10]	1,000	∼1,200	224×224	SRe^{2}L (Ours)

Table 1: Variants of ImageNet-1K training set with different configurations.

A.2 Squeezing Details

Data Augmentation.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

config	value
optimizer	SGD
base learning rate	0.2
weight decay	1e-4
optimizer momentum	0.9
batch size	256
learning rate schedule	cosine decay
training epoch	100
augmentation	RandomResizedCrop

(a) Tiny-ImageNet squeezing setting.

config	value
optimizer	AdamW
base learning rate	0.001
weight decay	0.01
optimizer momentum	β_1, β_2 = 0.9, 0.999
batch size	1,024
learning rate schedule	cosine decay
training epoch	300
augmentation	RandomResizedCrop

(b) ImageNet-1K validation setting.

config	value
α_{BN}	1.0
optimizer	Adam
base learning rate	0.1
weight decay	1e-4
optimizer momentum	β_1, β_2 = 0.5, 0.9
batch size	100
learning rate schedule	cosine decay
recovering iteration	1,000
augmentation	RandomResizedCrop

(c) Tiny-ImageNet recovering setting.

config	value
α_{BN}	0.01
optimizer	Adam
base learning rate	0.25
weight decay	1e-4
optimizer momentum	β_1, β_2 = 0.5, 0.9
batch size	100
learning rate schedule	cosine decay
recovering iteration	2,000
augmentation	RandomResizedCrop

(d) ImageNet-1K recovering setting.

Table 2: Parameter settings in three stages.

Table 2 in the main paper illustrates that the utilization of data augmentation techniques during the squeeze phase contributes to a decrease in the final accuracy of the data recovered. To summarize, the results on Tiny-ImageNet indicate that lengthening the training period and the application of data augmentation in the squeeze phase intensify the intricacy involved in data recovery from the compressed model.

Parallel conclusions are inferred from the compressed models for the ImageNet-1K dataset. For our experimental setup, we aimed to extract data from a pre-trained ResNet50 model with available *V1* and *V2* weights in the PyTorch model zoo. The results propose that the task of data extraction poses a greater challenge from the ResNet50 model equipped with *V2* weights as compared to the model incorporating *V1* weights. This can be attributed to the fact that models utilizing *V1* weights are trained employing a rudimentary recipe, whereas models with *V2* weights encompass numerous training enhancements, such as prolonged training and data augmentation, to achieve cutting-edge performance. These additional complexities impede the data recovery process. Therefore, the pre-trained models we employ for the recovery of ImageNet-1K images are those integrating *V1* weights from the PyTorch model zoo.

Hyper-parameter Setting.

• Tiny-ImageNet: We train modified ResNet-{18, 50} models on Tiny-ImageNet data with the parameter setting in Table 2a. The well-trained ResNet-{18, 50} models achieve Top-1 accuracy of {59.47%, 61.17%} under the 50 epoch training setting.

• ImageNet-1K: We use PyTorch off-the-shelf ResNet-{18, 50} with *V1* weights and Top-1 accuracy of {69.76%, 76.13%} as squeezed/condensed models. In the original training script [11], ResNet models are trained for 90 epochs with a SGD optimizer, learning rate of 0.1, momentum of 0.9 and weight decay of 1×10^{-4}.

A.3 Recovering Details

Regularization Terms. We conduct a multitude of ablation experiments under varying regularization term conditions, as illustrated in Table 3. The two image prior regularizers, L2 regularization and total variation (TV), are not anticipated to enhance validation accuracy as our primary focus

Ablation			Top-1 acc. (%)	
$\mathcal{R}_{TV}$	$\mathcal{R}_{\ell_2}$	Random Crop	Tiny-ImageNet	ImageNet-1K
✓	✓	✗	29.87	22.92
✓	✗	✗	29.92	23.15
✗	✓	✗	30.11	40.81
✗	✗	✗	30.30	40.37
✗	✗	✓	37.88	46.71

Table 3: Top-1 validation accuracy under ablation experiment settings. ResNet-18 is used in three stages with the relabling temperature $\tau = 20$.

is on information recovery rather than image smoothness. Consequently, we exclude these two regularization terms from our experiments.

Multi-crop Optimization. To offset the `RandomResizedCrop` operation applied to the training data during the model training phase, we incorporate a corresponding `RandomResizedCrop` augmentation on synthetic data. This implies that only a minor cropped region in the synthetic data undergoes an update in each iteration. Our experimentation reveals that our multi-crop optimization strategy facilitates a notable improvement in validation accuracy, as presented in Table 3. A comparative visualization with other non-crop settings in Fig. 1 shows multiple miniature regions enriched with categorical features spread across the entire image in the last columns (SRe^{2}L). Examples include multiple volcanic heads, shark bodies, bee fuzz, and mountain ridges. These multiple small feature regions populate the entire image, enhancing its expressiveness in terms of visualization. Therefore, the cropped regions on our synthetic images are not only more closely associated with the target categories but also more beneficial for model training.

Memory Consumption and Computational Cost. Regarding memory utilization, the memory accommodates a pre-trained model, reconstructed data, and the corresponding computational graph during the data recovery phase. Unlike the MTT approach, which necessitates all model states across all epochs during model training to align with the trajectory, our proposed methodology, SRe^{2}L, merely requires the statistical data from each Batch Normalization (BN) layer, stored within the condensed model, for image optimization. In terms of computational overhead, it is directly proportional to the number of recovery iterations. To establish a trade-off between performance and computational time, we enforce a recovery budget of 1k iterations for Tiny-ImageNet and 2k iterations for ImageNet-1K in ablation experiments. Our best accuracy, achieved on condensed data from 4k recovery iterations, is presented in Table 4 in the main paper.

Hyper-parameter Setting.

We calculate the total recovery loss $\ell_{total} = \underset{\mathcal{C}_{\text{syn}},|\mathcal{C}|}{\arg\min}\, \ell\left(\phi_{\boldsymbol{\theta}_\mathcal{T}}(\widetilde{\boldsymbol{x}}_{\text{syn}}), \boldsymbol{y}\right) + \alpha_{\text{BN}}\mathcal{R}_{\text{BN}}$ and update synthetic data with the parameter setting in Table 2c and Table 2d for Tiny-ImageNet and ImageNet-1K, respectively.

A.4 Relabeling & Validation Details

In this experiment, we utilize an architecture identical to that of a recovery model to provide soft labels as a teacher for synthesized images. We implement a fast knowledge distillation process for a duration of 300 epochs with a temperature setting of $\tau = 20$.

Hyper-parameter Setting. Regarding Tiny-ImageNet, we leverage the condensed data and the retargeted labels to train the validation model over a span of 100 epochs, with all other training parameters adhering to the condensing configurations outlined in Table 2a. In the case of ImageNet-1K, we train the validation model in accordance with the parameter configurations presented in Table 2b.

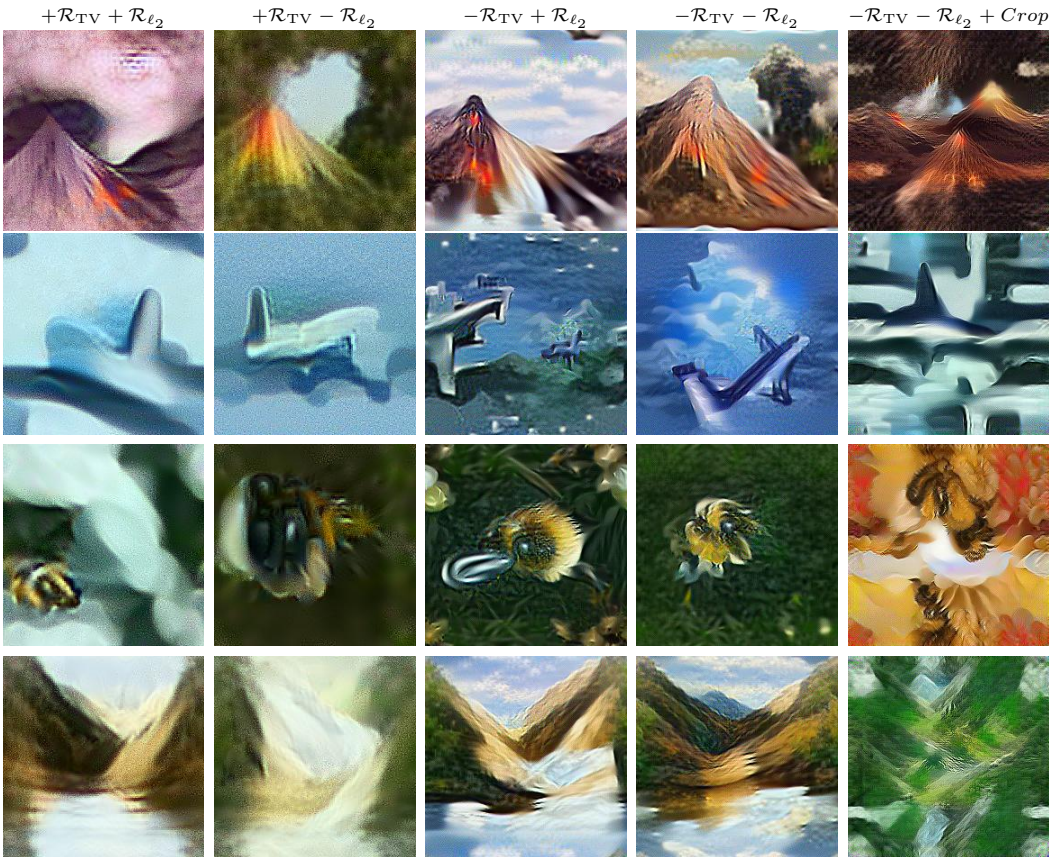


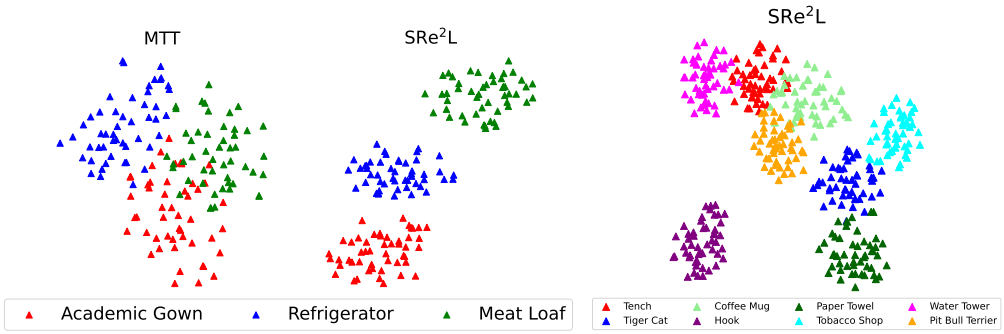
Figure 1: Distilled example visualization under various regularization terms and crop augmentation settings. Selected classes are {Volcano, Hammerhead Shark, Bee, Valley}.

B Low-Resolution Data (32×32)

Experiments were carried out on diminutive datasets such as MNIST and CIFAR. Intrinsically, these datasets encapsulate a limited quantum of information. Our method, which involves squeezing and subsequent recovering, inherently leads to information loss at each stage, thereby impeding the competitiveness of our results on these datasets. Nevertheless, our approach continues to demonstrate superior computational efficiency and enhanced processing speed when applied to these datasets.

C Feature Embedding Distribution

We feed the image data through a pretrained ResNet-18 model, subsequently extracting the feature embedding prior to the classification layer for the purpose of executing t-SNE [12] dimensionality reduction and visualization. Fig. 2a exhibits two distinct feature embedding distributions of synthetic Tiny-ImageNet data, sourced from 3 classes in MTT's and SRe^{2}L's condensed datasets, respectively. Relative to the distribution present in MTT, SRe^{2}L's synthetic data from differing classes displays a more dispersed pattern, whilst data from identical classes demonstrates a higher degree of clustering. This suggests that the data synthesized by SRe^{2}L boasts superior discriminability with respect to feature embedding distribution and can therefore be utilized to train models to attain superior performance. Fig. 2b portrays feature embedding distributions of SRe^{2}L's synthetic ImageNet data derived from 8 classes. Our synthetic ImageNet data also exemplifies exceptional clustering and discriminability attributes.



(a) Tiny-ImageNet (b) ImageNet-1K

Figure 2: Feature embedding distribution on synthetic data and real ImageNet-1K data. ResNet-18 is used as the feature embedding extractor.

D More Visualization of Synthetic Data

We provide more visualization comparisons on synthetic Tiny-ImageNet between MTT and SRe^{2}L in Fig. 3. Additionally, we furnish synthetic samples pertaining to ImageNet-1K in Fig. 4 and Fig. 5 for a more comprehensive understanding. It can be observed that our synthetic data has stronger semantic information than MTT with more object textures, shapes and details, which demonstrates the superior quality of our synthesized data.

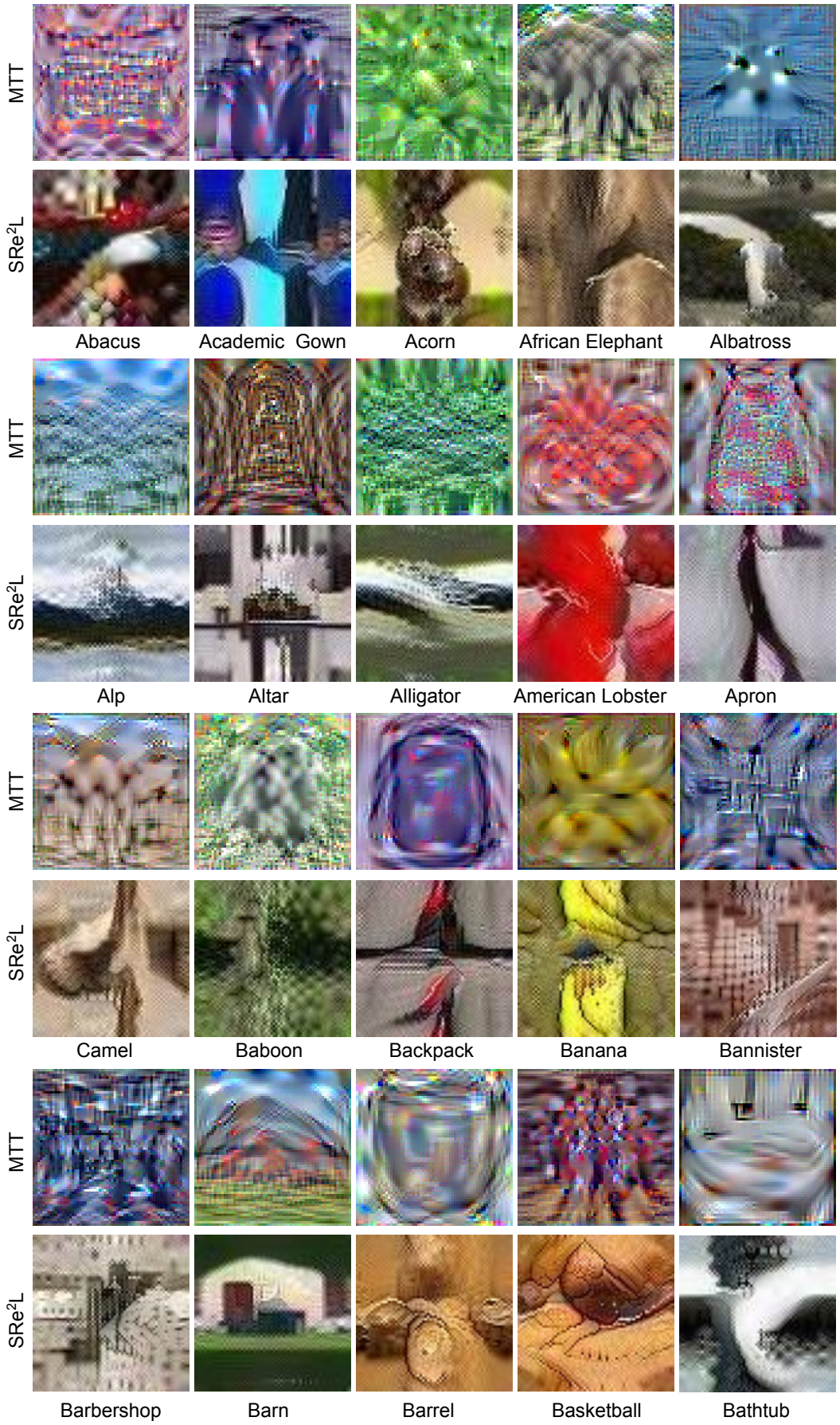


Figure 3: Synthetic Tiny-ImageNet data visualization from SRe^{2}L and MTT [3].

Figure 4: Synthetic ImageNet-1K data visualization from SRe^{2}L.

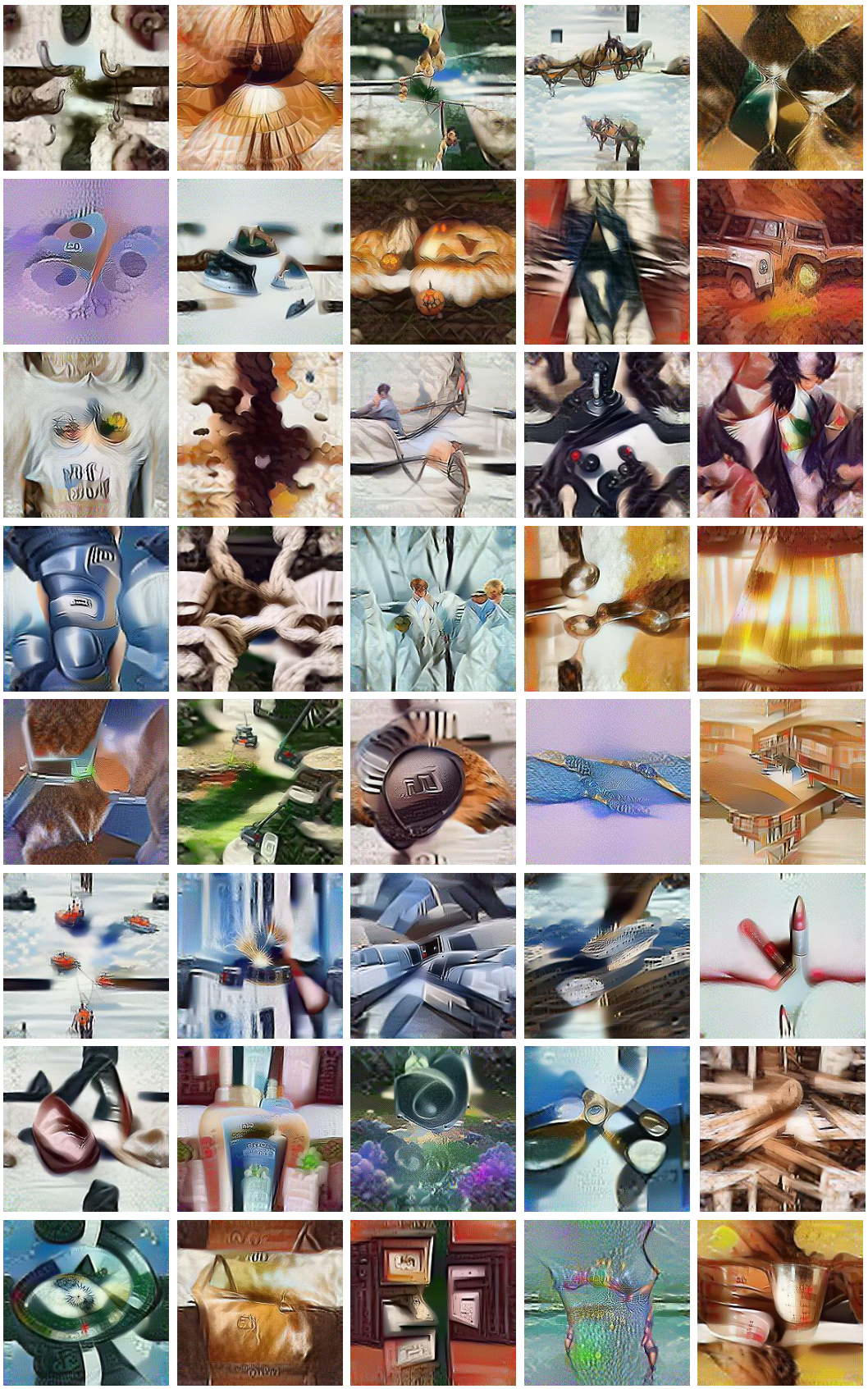


Figure 5: Synthetic ImageNet-1K data visualization from SRe^{2}L.

References

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

[2] Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from imagenet, March 2019. 1

[3] 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*, 2022. 1, 6

[4] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, 2020. 1

[5] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. In *Advances in Neural Information Processing Systems*, 2022. 1

[6] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, 2023. 1

[7] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *Proceedings of the 39th International Conference on Machine Learning*, 2022. 1

[8] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the CIFAR datasets. *arXiv preprint arXiv:1707.08819*, 2017. 1

[9] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. *arXiv preprint arXiv:2211.10586*, 2022. 1

[10] 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

[11] TorchVision maintainers and contributors. Torchvision image classification reference training scripts. https://github.com/pytorch/vision/tree/main/references/classification, 2016. 2

[12] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 4