A Appendix

B Dataset Distillation with Crucial Samples

In this section, we add more experiments showing the influence of discarding easy or hard examples on other datasets. The results are shown in Figure 5. There is a performance boost when easier samples are discarded at a small rate (red) while dropping the hardest ones can hurt the performances (green).

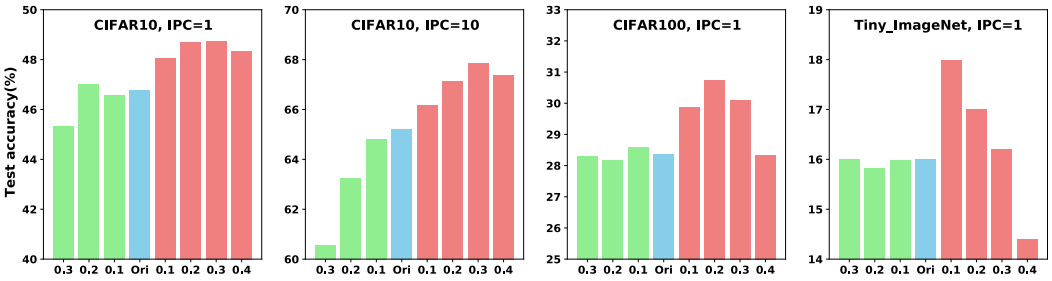


Figure 5: Accuracy performances on training networks under different situations. "Ori" indicates the original results. We first discard the 10%, 20%, 30% samples with the largest MSE loss in each batch to drop the hardest samples (green). The performance gets dropped compared to the original ones (blue). In contrast, when the easiest samples are discarded (red), the performances get a boost.

To investigate this phenomenon, we explore it from the perspective of data manifold and information.

Data Manifold: We show the distributions of synthetic images learned by the original baseline and baseline with discarding easy samples in Figure 1 (Middle) in the main text. Here we show all the distributions of baseline method, baseline with discarding hard samples, baseline with discarding easy samples, and baseline with the proposed ISA method in Figure 6. When the hard samples are discarded, it could be observed that the orange stars clusters together and the overlap with original dataset decreases. In contrast, the overlap grows when drop some easy samples, thereby depicting a better representation of the manifold. We also observe that compared to other cases, the application of ISA can help to make the stars more evenly distributed. This may explain why the generalization ability can be improved by ISA.

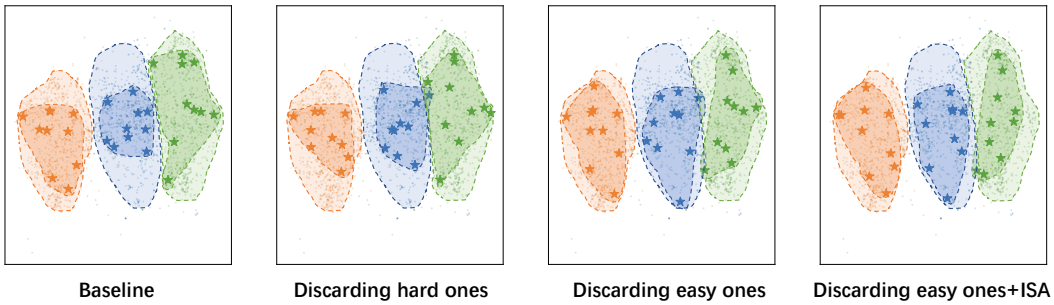


Figure 6: Distributions of synthetic images learned by different methods. The orange, blue, green points are the real images of three classes while the stars are the corresponding learned synthetic images. The orange stars clusters together and the overlap with original dataset decreases after discarding hard samples. In contrast, the overlap grows after discarding some easy samples, thereby depicting a better representation of the manifold. We also observe that compared to other cases, the application of ISA can help to make the stars more evenly distributed.

Information: To find out whether the harder samples have contained information in easier samples, we divide the target data into two splits: 80% samples that hold greater MSE loss in Eq. 4 and the left

20% easier ones. By treating the former as $X_{\mathcal{S}}$ and the latter as $X_{\mathcal{T}}$, we use the harder ones to make predictions for easier ones with Eq. 4. The MSE loss is 0.0661, as shown in Figure 1(Right). This loss indicates that the information in harder samples is enough for making precise predictions for easier samples. In contrast, when we use the 80% samples with smallest MSE loss to predict the left 20% hard ones, the loss is 0.1430. Therefore, harder samples cannot be replaced by simple samples.

Diversity: We also step further to compare the diversity of synthetic images with the recall value, which measures the expected likelihood of real samples against the synthetic manifold and is a commonly used metric in generative tasks for evaluating the diversity of generative model (Sajjadi et al., 2018). To be specific, in the generative model field, recall measures how much of a reference distribution can be generated by a part of a new distribution. Formally,

$$\text{recall} := \frac{1}{N}\sum_{i=1}^{N} 1_{R_i \in \text{manifold}(F_1,\ldots,F_M)}, \tag{8}$$

where N and M are the number of real and fake samples. $1_{(\cdot)}$ is the indicator function. F_i is the i-th fake sample while R_i is the i-th real sample. Manifolds are usually defined as:

$$\text{manifold}(R_1, ..., R_N) := \cup_{i=1}^{N} B(R_i, \text{NND}_k(R_i)), \tag{9}$$

where $B(x, r)$ is the sphere in $\mathbb{R}^D$ around x with radius r. $\text{NND}_k(R_i)$ denotes the distance from R_i to the k-th nearest neighbour among $\{R_i\}$ excluding itself.

To this end, the recall counts how many real samples occurs in the k-nearest neighbors of fake samples. We set $k = 5$. By treating the synthetic samples and original samples as fake and real samples respectively, we can calculate how many original samples can be recalled by generated images. A greater diversity in synthetic samples should recall more original samples. In other words, higher recall indicates greater diversity. With Eq. 8, we find that after incorporating the process of discarding easier samples in the outer loop, the recall value notably grows from 0.84 to 0.89. This improvement suggests an enhanced diversity in the synthetic samples with discarding easy samples.

C Algorithm Illustration.

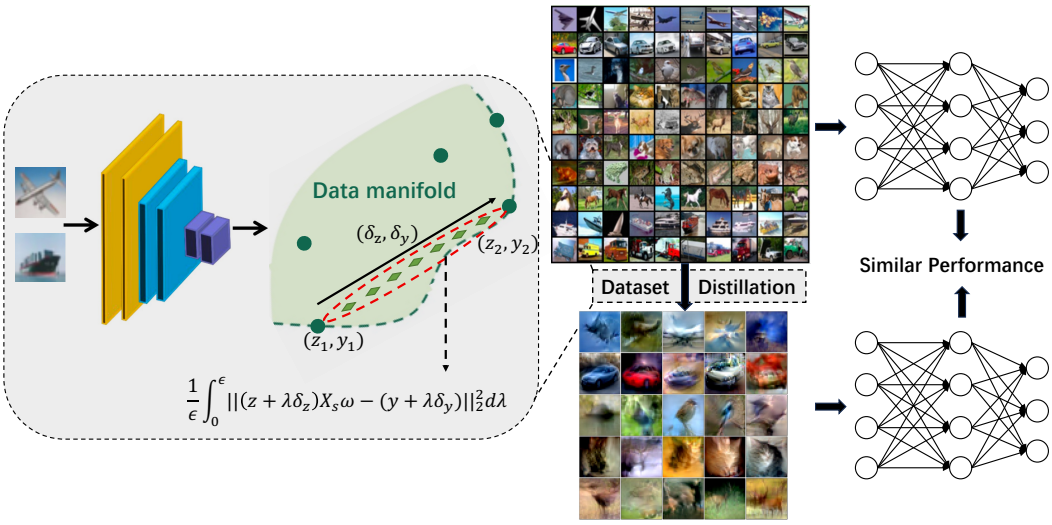


Figure 7: Dataset distillation with the assistance of crucial samples.

This paper introduces a dataset distillation algorithm based on crucial samples, which aims to distill a given labeled dataset into a smaller one so that a model trained on the small synthetic dataset can have a similar performance to the one trained on the original dataset, as shown in the right part of Figure 7. To achieve this goal, we first show that reducing redundancy in easy samples that are easy

to be represented by the generated samples and taking more crucial samples into consideration can be beneficial for improving the diversity of synthetic samples and better depicting the data manifold in the dataset distillation tasks. Based on this observation, we further develop an infinite semantic augmentation-based dataset distillation algorithm, which takes an infinite number of virtual crucial samples into consideration in the semantic space. Through detailed mathematical analysis, the joint contribution to training loss of all interpolated feature points is formed into an analytical closed-form solution of an integral that can be optimized with almost no extra computational cost. As shown in Figure 7, given two input samples $(\mathbf{x}_1, \mathbf{y}_1)$ and $(\mathbf{x}_2, \mathbf{y}_2)$, we first extract their features $\mathbf{z}_1, \mathbf{z}_2$ and then adopt the loss in this figure to take all the interpolated points between them into consideration. $\delta_z = \mathbf{z}_1 - \mathbf{z}_2, \delta_y = \mathbf{y}_1 - \mathbf{y}_2$.

The whole algorithm can also be found in Algorithm 1. It is established based on a state-of-the-art pipeline FRePo (Zhou et al., 2022), which implements the dataset distillation by: sampling a model uniformly from a model pool $\mathcal{M}$ (Line 3) and a target batch $(X_\mathcal{T}, Y_\mathcal{T})$ uniformly from the labeled dataset $\mathcal{T}$ (Line 4), then computing the meta-training loss $\mathcal{L}$ (Line 5) to update the distilled data $\mathcal{S}$ (Line 8) (outer loop) and training the model θ_i on $\mathcal{S}$ (Line 9) (inner loop). To conduct the crucial samples based dataset distillation, we add the crucial sample exploring procedure by finding the top $p * 100\%$ percent images with the greatest meta-training loss (Line 6). Based on these samples, we further take more virtual samples into consideration via the new meta-learning loss (Line 7,8).

D Infinite Semantic Augmentation

As the proposed Infinite Semantic Augmentation (ISA) takes an infinite number of virtual samples into consideration, one may be curious about whether the ISA will require more training steps for convergence. Figure 8 gives the answer. It suggests that there exists no big difference between the number of training steps for convergence of the baseline with that of method with ISA. Besides, the proposed method can achieve a better performance at the very early stages of training, indicating that the proposed method requires less time for a comparable performance. As for the training time cost, it is 2.5 hours (500,000 steps in total) under CIFAR10, IPC=10 setting while it is 2.4 hours for our baseline, indicating that the proposed module introduces negligible extra computational and time costs.

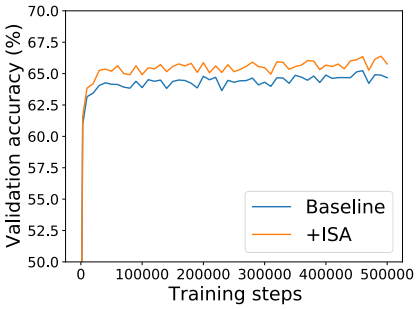


Figure 8: Validation accuracy during each training steps. It can be observed that adopting ISA will not require more training time for convergence.

The "single-step semantic augmentation" in Figure 3(c) indicates conducting a single-step mixup in feature space. As our method enables the augmentation of an infinite number of virtual samples in the semantic space by continuously interpolating between two target feature vectors, one may be curious about whether it is necessary to take infinite number of virtual samples into consideration. Therefore, in Figure 3(c), we conduct the single-step augmentation to take only one virtual samples between two samples into consideration during each step (mixup in feature-space). It can be found that our ISA can hold a better performance against vanilla MixUp and single-step semantic augmentation, indicating the superiority of the proposed ISA. We will update our paper to improve the readability.

E Implementation Details

All of our experiments are performed on a single NVIDIA A100 GPU with 80GB of GPU memory. We implement our method in JAX and reproduce previous methods using their officially released code. All the hyper-parameters are set following the released instructions. The training time for experiments on ImageNet including ImageNet-64, ImageNette, ImageWoof is around a week on a single A100, the same with the original ones. Other experiments only require several hours for training.

We also report the KRR predictors test accuracy using the feature extractor trained on the distilled data following FRePo (Zhou et al., 2022), which means obtaining the prediction with $K^{\theta}_{X_{\mathcal{T}}X_{\mathcal{S}}}(K^{\theta}_{X_{\mathcal{S}}X_{\mathcal{S}}} + \lambda I)^{-1}Y_{\mathcal{S}}$ in Eq. 4. With the KRR, we can achieve a higher test performance as shown in Table 6,7. The "ori" indicates training the neural networks with the distilled data and making predictions with the trained network, which is the default setting. However, we find the KRR predictor may fail to improve the performance when the distillation task is tough. For example, the test performance drops from 8.0% to 6.7% on the ImageNet dataset, as shown in Table 7.

Table 6: Distillation performance in term of KRR predicted test accuracy (%).

Method		CIFAR10			CIFAR100			TinyImageNet	
		1	10	50	1	10	50	1	10
FRePo	ori	46.8±0.7	65.5±0.4	71.7±0.2	28.7±0.1	42.5±0.2	44.3±0.2	15.4±0.3	25.4±0.2
	KRR	47.9±0.6	68.0±0.2	74.4±0.1	32.3±0.1	44.9±0.2	43.0±0.3	19.1±0.3	26.5±0.1
Ours	ori	48.4±0.4	67.2±0.4	73.8±0.0	31.2±0.2	46.4±0.5	**49.4±0.3**	19.8±0.1	27.0±0.3
	KRR	**50.5±0.7**	**69.0±0.4**	**75.6±0.1**	**38.0±0.1**	**48.4±0.4**	48.0±0.2	**23.4±0.4**	**28.1±0.2**

Table 7: Distillation performance in term of KRR predicted test accuracy (%) on ImageNet subsets.

Method		ImageNette (128x128)		ImageWoof (128x128)		ImageNet (64x64)	
		1	10	1	10	1	2
FRePo	ori	48.1±0.7	66.5±0.8	26.7±0.6	42.2±0.9	7.5±0.3	9.7±0.2
	KRR	**50.6±0.6**	67.1±0.7	31.3±0.9	43.5±0.8	7.2±0.2	9.5±0.2
Ours	ori	49.6±0.6	67.8±0.3	30.8±0.5	43.8±0.6	**8.0±0.2**	**10.7±0.1**
	KRR	48.5±0.6	**69.2±0.4**	**33.6±0.5**	**46.3±0.4**	6.7±0.2	7.6±1.0

F Experiments on Matching-based Methods

This paper mainly focuses on exploring what kind of target data is crucial for dataset distillation in the outer loop of the meta-learning-based methods based on the analysis of both the matching-based and meta-learning-based methods in the secondary paragraph in Introduction. With the exploration in Section 2.2, we introduce a selection+augmentation method that can be adopted during the outer loop of meta-learning-based methods. Therefore, apart from FRePo (Zhou et al., 2022)(the base model by default), we also combined the proposed modules with various state-of-the-art meta-learning-based methods including RFAD (Loo et al., 2022), FRePo (Zhou et al., 2022), RCIG (Loo et al., 2023) in our ablation studies. The results in Table 4 validate the effectiveness of the proposed method.

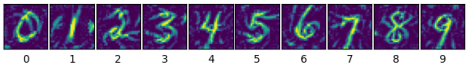
To further explore whether the proposed module can benefit for matching-based methods, we conduct experiments on classical DM (Zhao & Bilen, 2023) and MTT (Cazenavette et al., 2022) methods in Table 8. The performances are improved in most cases, indicating the effectiveness of the proposed approach.

Table 8: Test accuracies of applying our module to matching-based methods. * indicates the results are reproduced with the officially released codes.

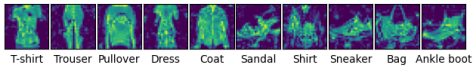
Methods	CIFAR10			Methods	CIFAR10		
	1	10	50		1	10	50
DM*	25.9±0.8	**48.9±0.6**	62.7±0.5	MTT*	46.3±0.8	65.2±0.5	71.6±0.2
+Ours	**26.5±0.6**	48.5±0.4	**62.9±0.2**	+Ours	**57.9±0.6**	**65.4±0.6**	**72.9±0.2**

G Visualization of Distilled Samples

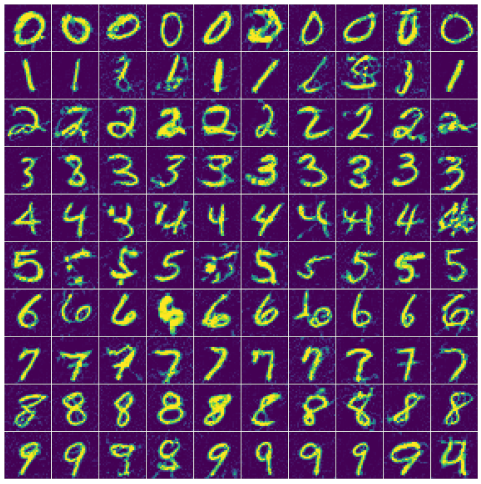
In this section, we show our distilled samples of various datasets under different IPCs.



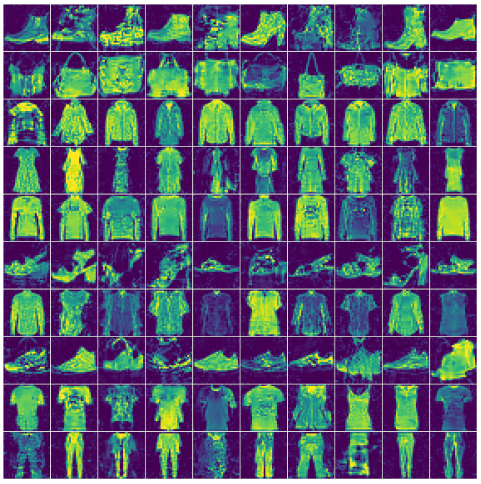
(a) MNIST, IPC=1



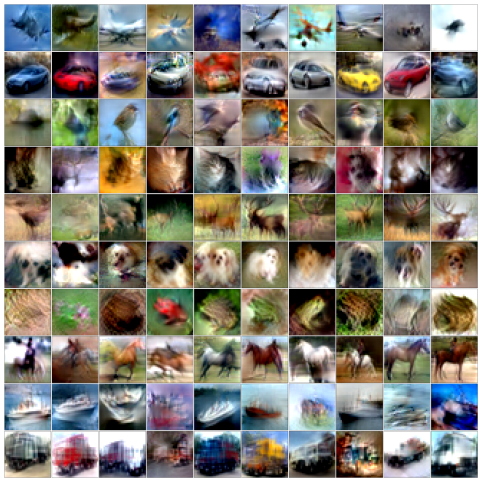
(b) Fashion-MNIST, IPC=1



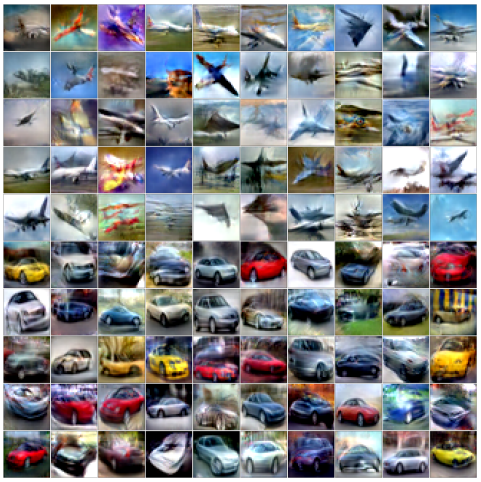
(c) MNIST, IPC=10



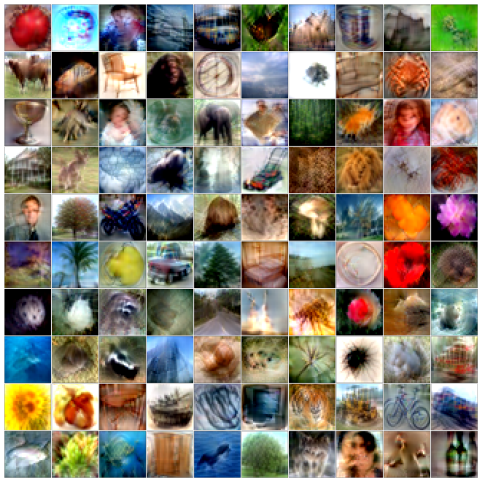
(d) Fashion-MNIST, IPC=10



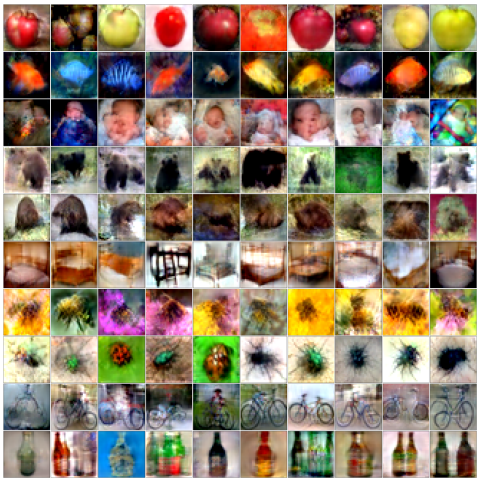
(e) CIFAR10, IPC=10



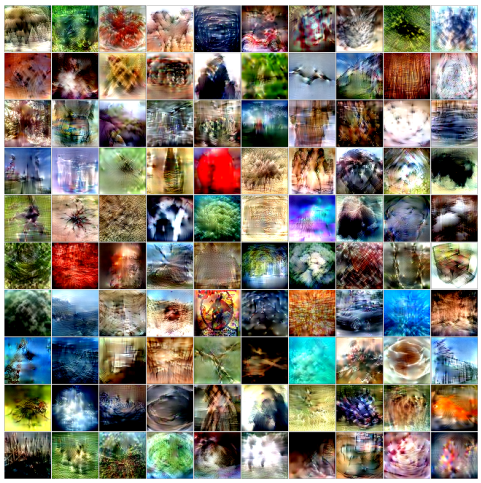
(f) CIFAR10, IPC=50



(g) CIFAR100, IPC=1

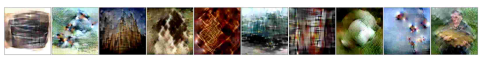


(h) CIFAR100, IPC=10



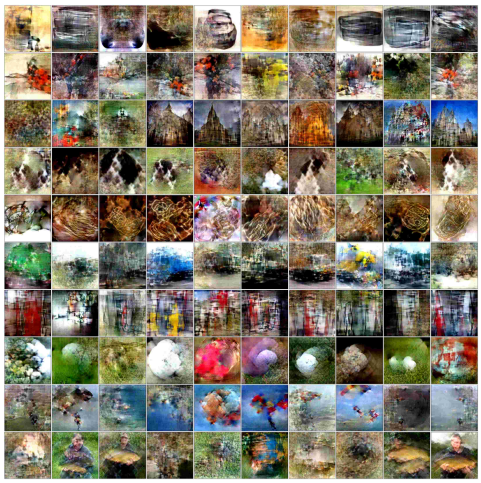
(i) TinyImageNet, IPC=1

(j) TinyImageNet, IPC=10



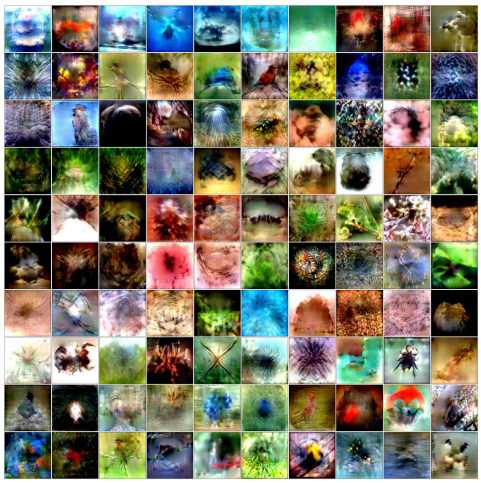
(k) ImageNette, IPC=1

(l) ImageWoof, IPC=1

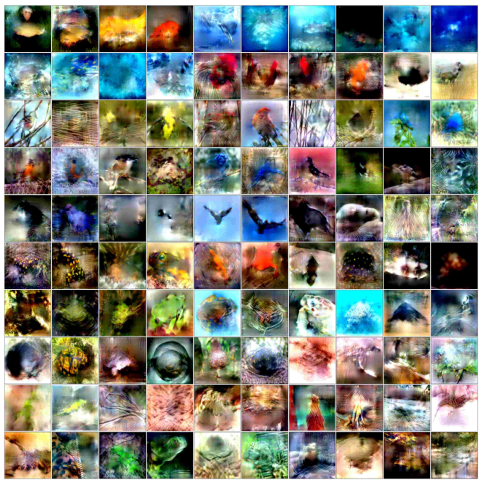


(m) ImageNette, IPC=10

(n) ImageWoof, IPC=10



(o) ImageNet (64x64), IPC=1



(p) ImageNet (64x64), IPC=2