[Supplementary Materials]
Disposable Transfer Learning for Source Task Unlearning

A. Experimental configurations

In this paper, we set different learning rates, the number of epochs, batch sizes, and the number of GPU processes for each training step.

Optimization We used cross-entropy loss for learning. For optimization, we used the SGD optimizer with `weight_decay=1e-4` and `momentum=0.9`. The learning rate is scheduled by a cosine annealing scheme with the initial learning rate η_0 described in the following paragraph.

Hyperparameters In pre-training of TL and knowledge disposal, we set the initial learning rate to $\eta_0 = 0.05$ for training 90 epochs with 4 GPUs. Because those stages are trained with the full size of the source dataset, it is acceptable to use larger batch size and more GPUs for faster training. Especially, for GC loss, we divide each mini-batch into $c = 4$ chunks so that each GPU calculates for a chunk. In fine-tuning of TL and piggyback learning (PL), the model is trained with an initial learning rate of $\eta_0 = 0.01$ for training 30 epochs with two GPUs. We reduced the batch size because the scale of training data is much smaller, but we matched the number of batches per GPU to 32.

For the TinyImageNet experiments, the dataset was downsampled to a resolution of 36×36 and random-cropped to 32×32 to be consistent with the input layer used for the CIFAR dataset, our target task. For pre-training source task, we used a batch size of 64, running on two GPUs, with an initial learning rate of 0.05 and trained over 90 epochs. For training target task, we employed a batch size of 32 across two GPUs with an initial learning rate of 0.01 and trained over 30 epochs. For the knowledge disposal stage, the settings were consistent with the experiments using CIFAR-100 as the source task: a batch size of 128, utilizing four GPUs, with an initial learning rate of 0.05. Concerning piggyback learning, the TinyImageNet piggyback was trained using the same setting as the knowledge disposal stage, whereas other datasets were piggybacked with a batch size of 128, using two GPUs, over 30 epochs, with a learning rate of 0.01.

Classification layers We assign classification layers per task which are linear layers that take the extracted features from the shared feature network, so the classification layer is independent between the preceding and the proceeding tasks.

Experiment	RAND	UNIF	NEG	GC
CIFAR-100 $\xrightarrow{DTL}$ CIFAR-10-1%	0.9000	0.9000	0.9750	0.3000
CIFAR-100 $\xrightarrow{DTL}$ STL-10-10%	0.6000	0.8500	0.9900	0.2600
TinyImageNet $\xrightarrow{DTL}$ CIFAR-100-10%	0.8750	0.9700	0.9875	0.2600

Table 5: λ values used for the experiments.

B. Details on the baseline methods

In the case of RAND, UNIF, NEG, and GC models, we set the best model with different values of λ for each unlearning loss in each experiment as in Table 5.

In addition, two additional baselines are reported in this material. The PRE and TGT model behaves similarly to the TL and TGT model, respectively. First, we estimated the PRE model, the output model of pre-training in TL. The PRE model is estimated to analyze the effect of pre-training and to compare the performance with the TL model. Likewise, we analyzed the SCR model, which is a randomly initialized model without any training. The TGT model is more suitable for comparing the performance gain with other unlearned models than the SCR model because the TGT model has sufficiently low target performance and it has little target knowledge.

C. Discussions on the DTL objective functions

C.1. Modified A-GEM for knowledge retaining

In Section 4.6 and Figure 6b, we compared the DTL performance of various retaining losses. Among them, TGT-A-GEM is a modified knowledge-retaining method based on the A-GEM method [1, 4] for continual learning. It prevents catastrophic forgetting of previous tasks by updating the gradient of the current task in a direction so as not to increase the loss of the previous tasks.

Drawing inspiration from the mechanism, we adopt the A-GEM algorithm to effectively retain the target knowledge. To prevent catastrophic forgetting of the target task, we added the constraint to our optimizing goal, Equa-

tion (1), as below. θ^t refers to the parameter state after t-th gradient descent update.

$$\textit{minimize} \quad \mathcal{L}_{dtl}(\theta^t) \tag{15}$$

$$\textit{subject to} \quad \mathcal{L}_{learn}(\theta^t) \leq \mathcal{L}_{learn}(\theta^{t-1}) \tag{16}$$

Equation (16) indicates the knowledge retaining loss has to be non-increasing as θ^t is updated by SGD, which implies learning the target data $\mathcal{D}_t$ not to be restrained by unlearning the source data $\mathcal{D}_s$. The difference between the original A-GEM algorithm and our TGT-A-GEM is the object of knowledge retaining and the kind of minimizing objective. In continual learning, the loss on previous tasks has to be non-increasing, but in our study, the loss on retaining data has to be non-increasing. Also, we minimize $\mathcal{L}_{dtl}$, a linearly interpolated loss, but A-GEM minimizes a single kind of loss calculated from the current task.

From now on, we follow Equation (4) in [4] to obtain the gradient update formula. We rephrase Equations (15) and (16) with respect to $\nabla\mathcal{L}$.

$$\text{minimize}_{\nabla\tilde{\mathcal{L}}} \quad \frac{1}{2}\|\nabla\mathcal{L}_{dtl} - \nabla\tilde{\mathcal{L}}\|_2^2 \tag{17}$$

$$\text{subject to} \quad \nabla\tilde{\mathcal{L}}^\top \nabla\mathcal{L}_{retain} \geq 0 \tag{18}$$

According to Equation (18), if $\nabla\tilde{\mathcal{L}}^\top \nabla\mathcal{L}_{retain}$ is non-negative, then $\nabla\tilde{\mathcal{L}} = \nabla\mathcal{L}_{dtl}$. If the inner product is negative, $\nabla\tilde{\mathcal{L}}$ is projected to $\nabla\mathcal{L}_{retain}$. The policy for gradient updating reflects that the gradient of DTL loss should be updated not to disturb the learning of the target task if there is a directional conflict between the learning gradient and the unlearning gradient. Finally, our solution for that problem is as Equation (19).

$$\nabla\tilde{\mathcal{L}} = \begin{cases} \nabla\mathcal{L}_{dtl}, & \text{if } \nabla\mathcal{L}_{dtl}^\top \nabla\mathcal{L}_{retain} \geq 0 \\ \nabla\mathcal{L}_{dtl} - \frac{\nabla\mathcal{L}_{dtl}^\top \nabla\mathcal{L}_{retain}}{\nabla\mathcal{L}_{retain}^\top \nabla\mathcal{L}_{retain}} \nabla\mathcal{L}_{retain}, & \text{otherwise} \end{cases} \tag{19}$$

We adopted the novel gradient update policy for DTL, but as shown in Figure 6b, it is found that the knowledge retaining performance of A-GEM on target data (TGT-A-GEM) is not clearly distinguished from naive knowledge retaining with the cross-entropy loss with the target data (TGT-CE). Fundamentally, due to the small scale of the target data, the models with retaining knowledge from the TL model by distillation outperform others.

C.2. Normalized gradient collision loss

We conduct an extra investigation on a variant of the GC loss where we eliminate the magnitude information from the gradient, named Normalized Gradient Collision (NGC) loss.

Definition C.1 (Normalized Gradient Collision loss) *Normalized GC loss is a variant of the GC loss (Equation* (7)*). It focuses on the angle between the loss and ignores the scale of gradients by minimizing only the cosine similarity of the gradient pairs.*

$$\mathcal{L}_{ngc}(\mathcal{D}, \theta) = \frac{1}{\binom{c}{2}} \sum_{m \neq n} \frac{\nabla\ell_m(\theta)^\top \nabla\ell_n(\theta)}{\|\nabla\ell_m(\theta)\|\|\nabla\ell_n(\theta)\|} \tag{20}$$

In practice, we have found that regularizing the grad norm as well as minimizing the variance, which corresponds to the GC model, results in better performance. We show an analysis of the NGC and GC loss in Appendix D.2.

C.3. KL-divergence cannot represent unlearning

In this section, we further discuss why simply minimizing the log-likelihood cannot lead to unlearning through a counter-example. In the main manuscript, we have defined the likelihood-minimizing objective function as $\mathcal{L}_{negative}$ in Equation (13).

The KL-divergence between a target distribution P and model distribution Q is as follows:

$$D_{KL}\left(P(y|x)||Q(y|x)\right) = \sum_{y \in \mathcal{Y}} P(y|x) \log \frac{P(y|x)}{Q(y|x)}. \tag{21}$$

It is seen that $D_{KL}(P||Q)$ diverges if there exists a sample $\hat{y} \in \mathcal{Y}$ such that $Q(y|x) \to 0$ and $P(y|x) > 0$ [2, 5]. Unlike the minimization of KL-divergence which results in distributional similarity, KL-divergence can be trivially maximized by degenerating the softmax score of a single sample. Moreover, a trivial maximization can be achieved by perturbing a few neurons on the uppermost layers, which is highly related to class prediction value, so it is much easier to increase the KL divergence with the intact feature extractor.

D. Extended experimental results

D.1. Full result of the main experiments

In Table 6, we provide the raw results of our CIFAR-100$\xrightarrow{DTL}$CIFAR-10-1%, CIFAR-100$\xrightarrow{DTL}$STL-10-10%, and TinyImageNet$\xrightarrow{DTL}$CIFAR-100-10% experiment, including source accuracy, target accuracy, and PL accuracy. It is visualized in various ways for motivating our work in Figure 2, or highlighting the relative performance change as shown in Table 1 and Table 3. From now on, we give a straightforward explanation of how we interpret and focus on the meaningful result based on the table.

The performance of the fine-tuning on different initializers, as plotted in Figure 2, is based on the PL accuracies of SCR (red), PRE(green), and TL (green) model. It motivates us to propose DTL, as claimed in the Introduction section, that the fine-tuned model also is a competitive representation model as the pre-trained model.

Model	Acc_s	$Acc_t\uparrow$	PL accuracy ↓		
			CIFAR-100-10%	STL-10-10%	SVHN-1%
SCR	1.00	8.53	25.75	39.58	22.40
PRE	71.54	11.41	68.20	63.45	60.82
TL	67.46	70.93	68.10	65.25	61.97
TGT	1.22	35.12	27.67	41.90	31.33
RAND	1.64	69.00	53.94	62.25	65.98
UNIF	2.74	71.41	55.06	64.08	69.63
NEG	0.02	71.17	60.92	65.16	71.10
GC	2.41	68.96	43.57	57.78	55.87

(a) CIFAR-100$\xrightarrow{DTL}$CIFAR-10-1%

Model	Acc_s	$Acc_t\uparrow$	PL accuracy ↓		
			CIFAR-100-10%	CIFAR-10-1%	SVHN-1%
SCR	1.09	10.00	25.75	35.12	22.40
PRE	71.54	10.56	68.20	70.93	60.82
TL	65.41	63.45	68.15	72.12	60.82
TGT	1.15	39.58	27.76	36.94	33.85
RAND	2.02	62.47	56.88	70.21	67.24
UNIF	7.03	63.04	55.67	69.57	67.22
NEG	0.02	60.86	57.75	67.97	70.47
GC	3.14	61.53	45.16	61.19	56.63

(b) CIFAR-100$\xrightarrow{DTL}$STL-10-10%

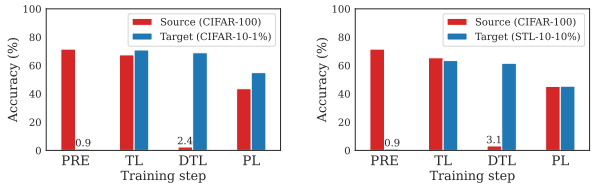
Model	Acc_s	$Acc_t\uparrow$	PL accuracy ↓		
			TinyImageNet-6%	STL-10-10%	CIFAR-10-1%
SCR	0.46	1.04	14.90	35.91	30.72
PRE	54.87	0.36	38.95	69.82	75.15
FT	38.06	55.99	39.12	67.71	72.08
TGT	0.41	25.16	17.16	45.44	40.82
RAND	1.22	53.79	29.20	65.81	67.40
UNIF	5.66	54.79	28.00	64.21	68.60
NEG	0.06	53.96	27.20	64.94	68.21
GC	5.46	54.35	26.40	62.83	64.18

(c) TinyImageNet$\xrightarrow{DTL}$CIFAR-100-10%

Table 6: The transition of source and target accuracy for each DTL stage and piggyback learning in CIFAR-100$\xrightarrow{DTL}$CIFAR-10-1%, CIFAR-100$\xrightarrow{DTL}$STL-10-10%, and TinyImageNet$\xrightarrow{DTL}$CIFAR-100-10% experiments.

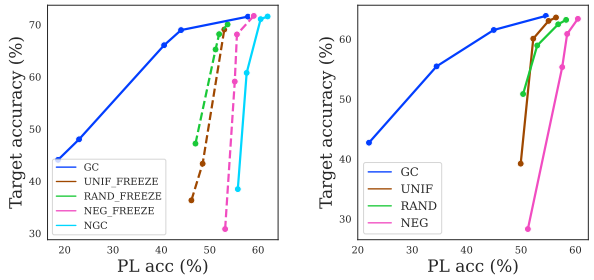
In Section 4.3, we emphasize the performance gain and penalty on the target task by reporting the relative performance in Table 1. It corresponds to the performance gap with TGT / TL models and our unlearned models in Table 6. PL accuracy is reported with the amount of performance changed from the TL model to our unlearned models. Comparing the relative performance of the GC model is a simpler way to evaluate its powerful DTL performance than comparing its absolute performance.

Also, we show the appropriateness of PL accuracy by reporting the absolute value of source accuracy and PL accuracy in Table 3. We mainly focused on the irrelevance of



(a) CIFAR-100 $\xrightarrow{DTL}$ CIFAR-10-1% (b) CIFAR-100 $\xrightarrow{DTL}$ STL-10-10%

Figure 8: Performance transition for each training step. The model is repeatedly fine-tuned from the SCR model in order of pre-training, fine-tuning, knowledge disposal, and piggyback learning. In other words, we can get the output models in the sequence of SCR $\rightarrow$ PRE $\rightarrow$ TL $\rightarrow$ DTL $\rightarrow$ PL.



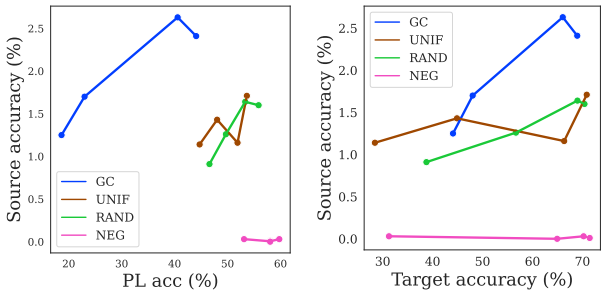
(a) CIFAR-100 $\xrightarrow{DTL}$ CIFAR-10-1% with unlearning variants (b) CIFAR-100 $\xrightarrow{DTL}$ STL-10-10% experiment

Figure 9: (a) The result of the variants of unlearning losses in CIFAR-100 $\xrightarrow{DTL}$ CIFAR-10-1% experiment, which is plotted in the same manner with Figure 6a. NGC loss model behavior is significantly different from GC loss model. Furthermore, baseline losses with frozen FC behave similarly as reported in the main manuscript. (b) The result of the same analysis with Figure 6a in CIFAR-100 $\xrightarrow{DTL}$ STL-10-10% experiment.

performance degradation (source accuracy) and knowledge disposal (PL accuracy). Note that the reported PL accuracy is the validation accuracy after training with a portion of the source data, CIFAR-100.

D.2. Variants of unlearning losses

As well as the unlearning baselines in the main manuscript, we also explore the variants in this section. NGC model, defined at Equation (20), and the variants of unlearning baselines are compared against our GC model in Figure 9a. The unlearning baselines, which are plotted in dashed lines, are unlearned by freezing the last classification layer of the source task and applying corresponding fooling losses. We freeze the classifier layer to test whether the baseline unlearning losses effects only the FC layer or not.



(a) Source accuracy vs PL accuracy. $\rho_s = -0.505$. (b) Source accuracy vs Target accuracy. $\rho_s = 0.199$.

Figure 10: The trade-off relationship of PL acc vs source accuracy and target accuracy vs source accuracy in CIFAR-100 $\xrightarrow{DTL}$ CIFAR-10-1% experiment with varying λ for each unlearning loss. ρ_s indicates the Spearman correlation coefficient.

Among the models in Figure 9a, we observed that the GC loss performs the best. The frozen variants (dashed lines) behave similarly to the result in Figure 6a. This implies that the baseline fooling losses only affect a few uppermost non-frozen layers. Interestingly, we have observed that the NGC loss performs worse than all other baselines unlearning loss, which shows that penalizing the gradient norm is also an important factor in gradient collision unlearning.

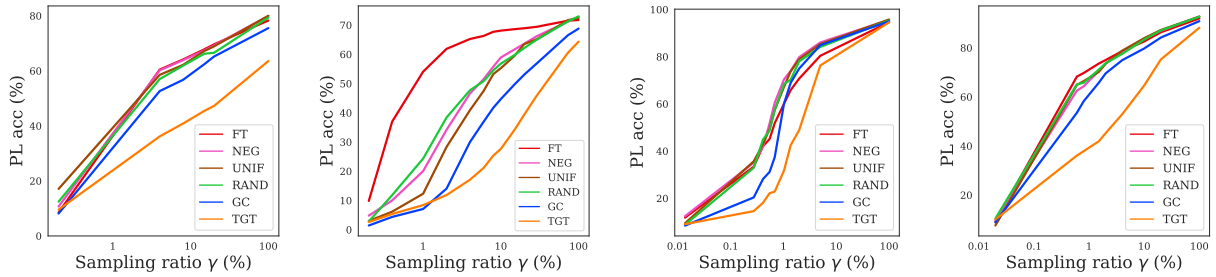
Another possible unlearning might be conducted by distilling to a randomly initialized model. However, it fails to transfer target task performance because the teacher lacks the target task knowledge, as demonstrated Acc_t in Table 9. This shows the target knowledge cannot be transferred by distillation. In a similar manner, other model compression methods such as pruning or quantization are not directly applicable for unlearning, as they are designed to best preserve the knowledge and cut down redundancy.

D.3. Risks and limitations of GC unlearning

A natural trade-off exists between knowledge disposal and the target task performance depending on the weight (λ) between knowledge retaining and unlearning loss. Notably, the proposed GC method demonstrated the most favorable trade-off (refer to 6a). We further inspected the behavior of GC unlearning by setting an extreme case when the source and target are identical (see Table 8). Increasing the weight on GC loss degrades the generalization performance while they equally achieve 100% train accuracy. This is because the GC loss acts as a regularization that favors over-fitted solutions. Additionally, we found that increasing λ results in higher loss curvature, which we measure by the trace of Hessian ($\mathrm{tr}(\mathbf{H}_\theta)$).

MIA strategy	SCRATCH		PRE		TL		TGT	
	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy
Softmax	49.61 $\pm$0.23	50.24 $\pm$0.13	67.96 $\pm$0.23	66.09 $\pm$0.25	63.71 $\pm$0.24	61.68 $\pm$0.28	49.86 $\pm$0.29	50.42 $\pm$0.20
Mentr.	50.06 $\pm$0.25	50.42 $\pm$0.17	70.14 $\pm$0.20	68.49 $\pm$0.19	67.33 $\pm$0.20	65.15 $\pm$0.18	50.12 $\pm$0.19	50.44 $\pm$0.13
Loss	50.00 $\pm$0.05	50.06 $\pm$0.04	69.93 $\pm$0.20	68.50 $\pm$0.19	67.25 $\pm$0.20	65.19 $\pm$0.18	49.97 $\pm$0.04	50.01 $\pm$0.00
Grad Norm	50.04 $\pm$0.25	50.40 $\pm$0.16	69.98 $\pm$0.20	68.60 $\pm$0.19	66.83 $\pm$0.21	65.16 $\pm$0.18	50.12 $\pm$0.19	50.44 $\pm$0.12
Adv. Dist	50.34 $\pm$0.21	50.66 $\pm$0.11	63.73 $\pm$0.18	64.34 $\pm$0.05	63.03 $\pm$0.19	63.63 $\pm$0.14	50.36 $\pm$0.23	50.70 $\pm$0.17
†Grad w	49.91 $\pm$0.39	50.46 $\pm$0.26	70.66 $\pm$0.52	68.75 $\pm$0.31	67.55 $\pm$0.46	65.33 $\pm$0.25	49.74 $\pm$0.34	50.41 $\pm$0.16
†Grad x	49.96 $\pm$0.39	50.55 $\pm$0.20	71.00 $\pm$0.43	68.72 $\pm$0.30	67.48 $\pm$0.35	65.09 $\pm$0.24	50.15 $\pm$0.44	50.67 $\pm$0.25
†Int. Outs	49.80 $\pm$0.45	50.46 $\pm$0.27	52.17 $\pm$0.39	51.93 $\pm$0.36	51.67 $\pm$0.53	51.65 $\pm$0.42	49.88 $\pm$0.33	50.48 $\pm$0.20
†WB	49.87 $\pm$0.27	50.46 $\pm$0.16	70.82 $\pm$0.43	68.60 $\pm$0.30	67.88 $\pm$0.38	65.29 $\pm$0.26	50.08 $\pm$0.48	50.59 $\pm$0.33
MIA strategy	GC		RAND		NEG		UNIF	
	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy
Softmax	50.76 $\pm$0.22	50.93 $\pm$0.16	51.51 $\pm$0.32	51.47 $\pm$0.22	50.91 $\pm$0.26	50.96 $\pm$0.22	50.65 $\pm$0.28	50.79 $\pm$0.21
Mentr.	50.23 $\pm$0.29	50.47 $\pm$0.19	51.41 $\pm$0.19	51.41 $\pm$0.14	52.06 $\pm$0.26	51.90 $\pm$0.19	53.88 $\pm$0.13	53.18 $\pm$0.11
Loss	50.23 $\pm$0.29	50.47 $\pm$0.19	51.42 $\pm$0.19	51.42 $\pm$0.14	56.46 $\pm$0.24	55.01 $\pm$0.18	53.88 $\pm$0.13	53.18 $\pm$0.11
Grad Norm	49.90 $\pm$0.29	50.34 $\pm$0.14	49.49 $\pm$0.23	50.20 $\pm$0.07	52.65 $\pm$0.22	52.43 $\pm$0.24	49.23 $\pm$0.22	50.29 $\pm$0.09
Adv. Dist	50.05 $\pm$0.06	50.11 $\pm$0.05	50.22 $\pm$0.06	50.23 $\pm$0.06	49.99 $\pm$0.00	50.00 $\pm$0.00	50.39 $\pm$0.07	50.40 $\pm$0.08
†Grad w	50.23 $\pm$0.47	50.69 $\pm$0.29	50.63 $\pm$0.42	50.90 $\pm$0.31	52.07 $\pm$0.54	51.91 $\pm$0.41	50.83 $\pm$0.48	51.10 $\pm$0.35
†Grad x	49.84 $\pm$0.51	50.45 $\pm$0.29	50.12 $\pm$0.47	50.58 $\pm$0.20	50.40 $\pm$0.55	50.74 $\pm$0.30	50.21 $\pm$0.60	50.64 $\pm$0.38
†Int. Outs	49.89 $\pm$0.59	50.52 $\pm$0.31	50.15 $\pm$0.62	50.71 $\pm$0.38	50.86 $\pm$0.62	51.10 $\pm$0.42	49.90 $\pm$0.51	50.52 $\pm$0.28
†WB	49.99 $\pm$0.42	50.52 $\pm$0.23	51.46 $\pm$0.64	51.49 $\pm$0.42	59.17 $\pm$0.51	56.93 $\pm$0.39	53.59 $\pm$0.52	53.14 $\pm$0.40

Table 7: The success rate of MIAs of models in CIFAR-100→CIFAR-10-1%. MIA strategies used are [3, 6, 7]. † involves training an attacker model.



(a) STL10 PL accuracy in CIFAR-100$\xrightarrow{DTL}$CIFAR-10-1%.

(b) CIFAR-100 PL accuracy in CIFAR-100$\xrightarrow{DTL}$STL-10-10%.

(c) SVHN PL accuracy in CIFAR-100$\xrightarrow{DTL}$STL-10-10%.

(d) CIFAR-10 PL accuracy in CIFAR-100$\xrightarrow{DTL}$STL-10-10%.

Figure 11: PL accuracy for different sampling ratio γ

Loss	λ	Train acc.	Test acc.	tr($\mathbf{H}_\theta$)
CE	0.00	100.00	76.13	66.2
CE+GC	0.10	100.00	72.18	226.2
CE+GC	0.15	100.00	71.12	314.4

Table 8: GC loss training when source task and target task are identical (CIFAR-100). The norm of the curvature is measured by the trace of the Hessian matrix.

Method	ΔAcc_t vs TGT ↑	ΔAcc_t vs TL ↑	ΔAcc_{pl} vs TL ↓		
			CIFAR100-10%	STL10-10%	SVHN-1%
R18→R18	+3.98	-31.83	-39.31	-10.32	-34.71
R50→R18	+3.88	-31.93	-39.06	-10.68	-32.59
Reference	35.12	70.93	68.15	53.28	61.97

Table 9: DTL using knowledge distillation. R18 and R50 indicate ResNet-18/50.

D.4. The source accuracy cannot verify the knowledge disposal

In Section 4.5, we have briefly discussed why the source accuracy of the unlearned model cannot verify the success of knowledge disposal with Table 3.

Moreover, there are additional reasons why the source accuracy cannot measure the degree of unlearning. Primar-

ily, our work is motivated by the situation that the TL model is susceptible to be adapted to various tasks except for the target task. As a result, the source accuracy can be ignored for model evaluation in the first place.

Also, the source accuracy cannot represent the conformability of other tasks. We demonstrated the trade-off of the PL accuracy vs the source accuracy (Figure 10a) and the target accuracy vs the source accuracy (Figure 10b). Those are plotted with the same data for Figure 6a, in which we evaluated the performance with varying the unlearning loss and λ in CIFAR-100$\xrightarrow{DTL}$CIFAR-10-1% experiment. As shown in Figure 10a, there the source accuracy has a low effect on the PL accuracy. Particularly, source accuracy is even negatively correlated with the PL accuracy because the Spearman correlation coefficient of them is $\rho_s = -0.505$.

As mentioned in Table 3, the source accuracy is nearly uncorrelated to the target accuracy. It is found that except GC models, the source accuracy of every model is lower than 2% with various ranges of target accuracy in Figure 10b and the Spearman correlation coefficient measured on the target accuracy and source accuracy is $\rho = 0.199$.

D.5. Complete results

In Figure 11a, PL accuracy versus different sampling ratios on the STL-10 dataset is reported, whereas there are other analyses on different PL datasets after CIFAR-100$\xrightarrow{DTL}$CIFAR-10-1% in Figure 5.

In addition, we also conducted the same analysis with Figure 5 and Figure 6a in the CIFAR-100$\xrightarrow{DTL}$STL-10-10% experiment. Refer to Figures 11b to 11d and Figure 9b, respectively.

D.6. Extended results on the membership inference attacks (MIAs)

In Table 2, we have reported the success rate of various white-box MIAs – Adv. Dist, Grad w, Grad x, and WB – for our unlearning models. Furthermore, in Table 7, we provide extended results with more models and additional attack methods. We adapted the implementation of [3] and followed experimental details. Specifically, the two measures, *AUROC* and *accuracy*, are measured by varying the threshold for binary classification of membership inference. AUROC represents the trade-off of true positive ratio (TPR) and false positive ratio (FPR) and accuracy is the best accuracy among various thresholding. Also, the reported values are averaged success rates and corresponding standard deviations of 20 runs.

Here, we provide details on each attack method. First, we attacked the models with various black-box methods. The Softmax response (Softmax) method is the most naive attack method, which attacks the model with the softmax output from the assumption that the predicted output from training data will be more confident. Modified entropy (Mentr.) is a variant of the Softmax method because it measures the output's uncertainty and it determines the sample with low uncertainty to the training sample. The Loss (Loss) method finds out the training sample by measuring the loss function and decides the sample with a lower loss to the training data.

Also, we applied the white-box methods to our models. The gradient norm (Grad Norm) method calculates the ℓ_2-norm of the gradient with respect to the model parameter and figures out the sample with a smaller norm to be a training sample. [3] proposes the adversarial distance (Adv. Dist) to measure the amount of perturbation of an example so that the model wrongly predicts the class of the sample in a white-box manner. [7] proposes a white-box attack method based on the gradient of the loss with respect to model weight (Grad w) and input (Grad x). The authors claim that the larger norm of both gradients indicates the sample is not a member of the training set. Though excluded in the main manuscript, the intermediate outputs (Int.Outs) method attacks a target model with the outputs of the final two layers as introduced in [7]. White-box method (WB) [6] takes intermediate feature and gradient for MIA.

References

[1] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019.

[2] Ankush Ganguly and Samuel W. F. Earp. An introduction to variational inference. *CoRR*, abs/2108.13083, 2021.

[3] Ganesh Del Grosso, Hamid Jalalzai, Georg Pichler, Catuscia Palamidessi, and Pablo Piantanida. Leveraging adversarial examples to quantify membership information leakage. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10389–10399, 2022.

[4] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NIPS*, 2017.

[5] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013.

[6] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 739–753. IEEE, 2019.

[7] S. Rezaei and X. Liu. On the difficulty of membership inference attacks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7888–7896, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society.