This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

# MAP: MAsk-Pruning for Source-Free Model Intellectual Property Protection

Boyang Peng[1*], Sanqing Qu[1*], Yong Wu[1], Tianpei Zou[1], Lianghua He[1],
Alois Knoll[2], Guang Chen[1†], Changjun Jiang[1]
[1]Tongji University, [2] Technical University of Munich

## Abstract

*Deep learning has achieved remarkable progress in various applications, heightening the importance of safeguarding the intellectual property (IP) of well-trained models. It entails not only authorizing usage but also ensuring the deployment of models in authorized data domains, i.e., making models exclusive to certain target domains. Previous methods necessitate concurrent access to source training data and target unauthorized data when performing IP protection, making them risky and inefficient for decentralized private data. In this paper, we target a practical setting where only a well-trained source model is available and investigate how we can realize IP protection. To achieve this, we propose a novel MAsk Pruning (MAP) framework. MAP stems from an intuitive hypothesis, i.e., there are target-related parameters in a well-trained model, locating and pruning them is the key to IP protection. Technically, MAP freezes the source model and learns a target-specific binary mask to prevent unauthorized data usage while minimizing performance degradation on authorized data. Moreover, we introduce a new metric aimed at achieving a better balance between source and target performance degradation. To verify the effectiveness and versatility, we have evaluated MAP in a variety of scenarios, including vanilla source-available, practical source-free, and challenging data-free. Extensive experiments indicate that MAP yields new state-of-the-art performance. Code will be available at https://github.com/ispc-lab/MAP.*

## 1. Introduction

With the growing popularity of deep learning in various applications (such as autonomous driving, medical robotics, virtual reality, etc), the commercial significance of this technology has soared. However, obtaining well-trained models is a resource-intensive process. It requires considerable time, labor, and substantial investment in terms of dedicated architecture design [12, 18], large-scale high-quality data [10, 21], and expensive computational resources [58]. Consequently, safeguarding the intellectual property (IP) of well-trained models has received significant concern.

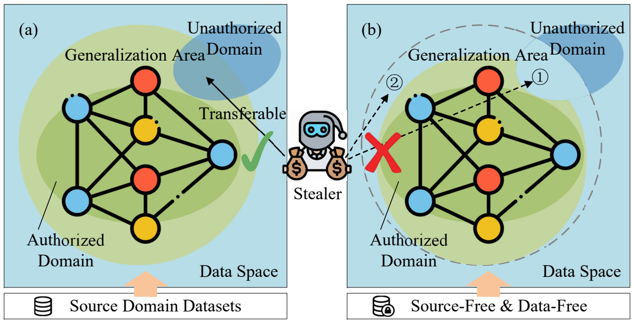

Figure 1. An illustration of model IP protection in source-free and data-free situations. (**a**) The original model is well-trained in the authorized (source) domain, with a wide generalization area that allows illegal access to the model through unauthorized (target) domains. (**b**) Two methods are shown: (1) Source-free IP protection, which removes an unauthorized domain from the generalization area without using source datasets; and (2) Data-Free IP protection, which cannot access any datasets but reduces the generalization area, preventing illegal knowledge transfer.

Previous studies on IP protection mainly focus on ownership verification [3, 23, 47] and usage authorization [16, 38], i.e., verifying who owns the model and authorizing who has permission to use it. Despite some effectiveness, these methods are vulnerable to fine-tuning or re-training. Moreover, authorized users retain the freedom to apply the model to any data without restrictions. Consequently, they effortlessly transfer high-performance models to similar tasks, leading to hidden infringement. Therefore, comprehensive IP protection requires a thorough consideration of applicability authorization. It entails not only authorizing usage but also preventing the usage of unauthorized data.

The primary challenge lies in the fact that the generalization region of well-trained models typically encompasses some unauthorized domains (as depicted in Fig. 1 (a)). It arises from the models' innate ability to capture domain-invariant features, thereby leading to potential applicability IP infringements. An intuitive solution is to make the generalization bound of models more explicit

[*] Equal Contribution
[†] Corresponding author: guangchen@tongji.edu.cn

and narrower, i.e., optimizing models to prioritize domain-dependent features and confining their applicability exclusively to authorized domains. To achieve this, NTL [49] first remolds the methodology in domain adaptation with an opposite objective. It amplifies the maximum mean difference (MMD) between the source (authorized) and target (unauthorized) domains, thereby effectively constraining the generalization scope of the models. Different from NTL, CUTI-domain [50] constructs middle domains with combined source and target features, which then act as barriers to block illegal transfers from authorized to unauthorized domains. Regardless of promising results, these methods require concurrent access to both source and target data when performing IP protection, rendering them unsuitable for decentralized private data scenarios. Moreover, they typically demand retraining from scratch to restrict the generalization boundary, which is highly inefficient since we may not have prior knowledge of all unauthorized data at the outset, leading to substantial resource waste.

In this paper, we target a practical but challenging setting where a well-trained source model is provided instead of source raw data, and investigate how we can realize source-free IP protection. To achieve this, we first introduce our *Inverse Transfer Parameter Hypothesis* inspired by the lottery ticket hypothesis [13]. We argue that well-trained models contain parameters exclusively associated with specific domains. Through deliberate pruning of these parameters, we effectively eliminate the generalization capability to these domains while minimizing the impact on other domains. To materialize this idea, we propose a novel MAsk Pruning (**MAP**) framework. MAP freezes the source model and learns a target-specific binary mask to prevent unauthorized data usage while minimizing performance degradation on authorized data. For a fair comparison, we first compare our MAP framework with existing methods when source data is available, denoted as SA-MAP. Subsequently, we evaluate MAP in source-free situations. Inspired by data-free knowledge distillation, we synthesize pseudo-source samples and amalgamate them with target data to train a target-specific mask for safeguarding the source model. This solution is denoted as SF-MAP. Moreover, we take a step further and explore a more challenging data-free setting, where both source and target data are unavailable. Building upon SF-MAP, we introduce a diversity generator for synthesizing auxiliary domains with diverse style features to mimic unavailable target data. We refer to this solution as DF-MAP. For performance evaluation, current methods only focus on performance drop on target (unauthorized) domain, but ignore the preservation of source domain performance. To address this, we introduce a new metric Source & Target Drop (*ST-D*) to fill this gap. We have conducted extensive experiments on several datasets, the results demonstrate the effectiveness of our MAP framework. The key contributions are summarized as follows:

- To the best of our knowledge, we are the first to exploit and achieve source-free and data-free model IP protection settings. These settings consolidate the prevailing requirements for both model IP and data privacy protection.
- We propose a novel and versatile MAsking Pruning (**MAP**) framework for model IP protection. MAP stems from our *Inverse Transfer Parameter Hypothesis*, i.e., well-trained models contain parameters exclusively associated with specific domains, pruning these parameters would assist us in model IP protection.
- Extensive experiments on several datasets, ranging from source-available, source-free, to data-free settings, have verified and demonstrated the effectiveness of our MAP framework. Moreover, we introduce a new metric for thorough performance evaluation.

## 2. Related Work

**Model Intellectual Property Protection.** To gain improper benefits and collect private information in the model, some individuals have developed attack methods, such as the inference attack [2, 31, 54], model inversion attack [14, 45, 57], adversarial example attack [34, 55] and others [1, 33]. Therefore, the protection of model intellectual property rights has become important. Recent research has focused on ownership verification and usage authorization to preserve model intellectual property [49]. Traditional ownership verification methods [25, 56] employ watermarks to establish ownership by comparing results with and without watermarks. However, these techniques are also susceptible to certain watermark removal [5, 17] techniques. Usage authorization typically involves encrypting the entire or a portion of the network using a pre-set private key for access control purposes [16, 38].

As a usage authorization method, NTL [49] builds an estimator with the characteristic kernel from Reproduction Kernel Hilbert Spaces (RKHSs) to approximate the Maximum Mean Discrepancy (MMD) between two distributions to achieve the effect of reducing generalization to a certain domain. According to [50], CUTI-Domain generates a middle domain with source style and target semantic attributes to limit generalization region on both the middle and target domains. [48] enhances network performance by defining a divergence ball around the training distribution, covering neighboring distributions, and maximizing model risk on all domains except the training domain. However, the privacy protection policy results in difficulties in getting user source domain database data, which disables the above methods. To address this challenge, in this paper, we propose source-free and target-free model IP protection tasks.

**Unstructured Parameter Pruning.** Neural network pruning reduces redundant parameters in the model to alleviate storage pressure. It is done in two ways: unstruc-

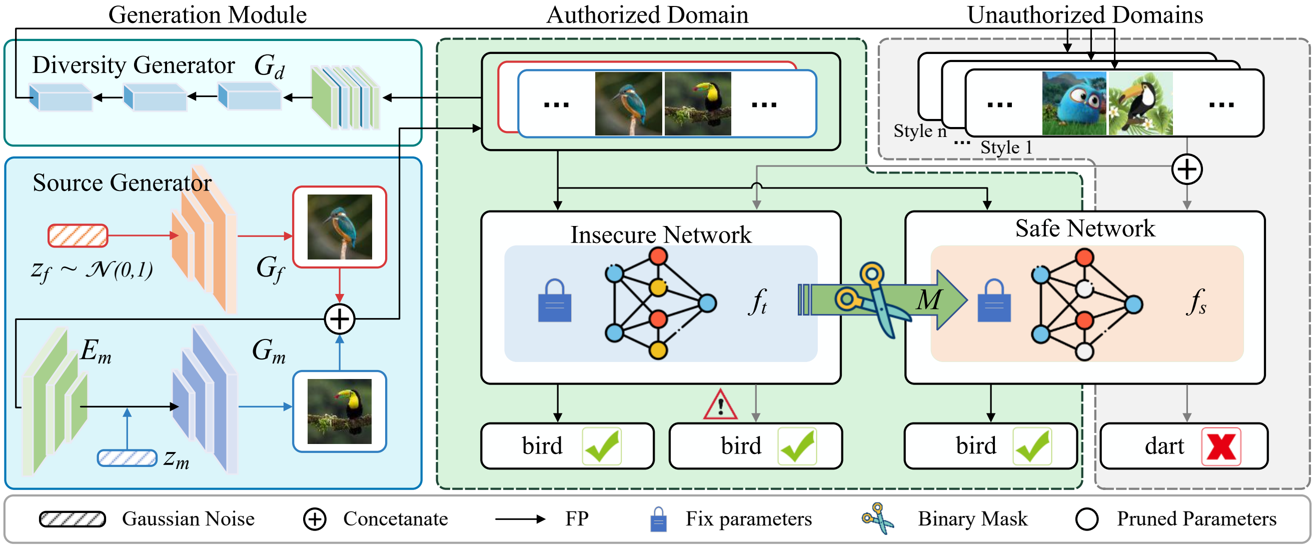

Figure 2. Overall architecture of MAP. Please note that this architecture presents the complete DF-MAP, from which SA-MAP and SF-MAP are derived. **(a)** The Generation Module, displayed in the left part, consists of three generators. The *Diversity Generator* ($G_d$) synthesizes auxiliary samples to generate neighbor domains with multiple style features. The *Fresh Generator* ($G_f$) generates synthetic novel featured samples, while *Memory Generator* ($G_m$) replays samples with features from previous images. In SF-MAP, the *Diversity Generator* ($G_d$) is removed, and existing target domain data is utilized for training. In SA-MAP, the entire Generation Module is eliminated, and existing source domain data is further leveraged, as detailed in the supplementary material. **(b)** The right part illustrates the mask-pruning process. A well-trained original source network $f_s$ distills knowledge into the target network $f_t$, which shares the same architecture. We initialize and fix them with the same checkpoint, then update a *Learnable Binary Mask* ($M$) with consistency loss calculated from synthetic samples. The MAP limits a target domain generalization region while retaining source domain performance, leading to a beneficial outcome.

tured or structured [19]. Structured pruning removes filters, while unstructured pruning removes partial weights, resulting in fine-grained sparsity. Unstructured pruning is more effective, but sparse tensor computations save runtime, and compressed sparse row forms add overhead [53].

There are several advanced methods for unstructure pruning. For example, [35] proposes a method that builds upon the concepts of network quantization and pruning, which enhances the network performance for a new task by utilizing binary masks that are either applied to unmodified weights on an existing network. As the basis of our hypothesis, the lottery hypothesis [13] illustrates that a randomly initialized, dense neural network has a sub-network that matches the test accuracy of the original network after at most the same number of iterations trained independently.

**Source-Free Domain Adaptation.** Domain adaptation addresses domain shifts by learning domain-invariant features between source and target domains [51]. In terms of source-free learning, our work is similar to source-free domain adaptation, which is getting more attention due to the data privacy policy [28]. [6] first considers prevention access to the source data in domain adaptation and then adjusts the source pre-trained classifier on all test data. Several approaches are used to apply the source classifier to unlabeled target domains [8, 46], and the current source-free adaptation paradigm does not exploit target labels by default [26, 29, 42]. Some schemes adopt the paradigm based on data generation [24, 30, 39], while others adopt the paradigm based on feature clustering [26, 27, 40, 41]. Our technique follows the former paradigm, which synthesizes a pseudo-source domain with prior information.

## 3. Methodology

In this paper, we designate the source domain as the authorized domain and the target domain as the unauthorized domain. Firstly, Section 3.1 defines the problem and our *Inverse Transfer Parameter Hypothesis*. Then Section 3.2, Section 3.3, and Section 3.4 present our MAsk-Pruning (MAP) framework in source-available, source-free, and data-free situations, respectively.

### 3.1. Problem Definition

Formally, we consider a source network $f_s : \mathcal{X}_s \to \mathcal{Y}_s$ trained on the source domain $\mathcal{D}_s = \{(x_s, y_s)||x_s \sim \mathcal{P}^s_{\mathcal{X}}, y_s \sim \mathcal{P}^s_{\mathcal{Y}}\}$, a target network $f_t : \mathcal{X}_t \to \mathcal{Y}_t$, and target domain $\mathcal{D}_t = \{(x_t, y_t)||x_t \sim \mathcal{P}^t_{\mathcal{X}}, y_t \sim \mathcal{P}^t_{\mathcal{Y}}\}$. $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{P}_{\mathcal{Y}}$ are the distribution of $\mathcal{X}$ and $\mathcal{Y}$, respectively. The goal of ***source-available IP protection*** is to fine-tune $f_t$ while minimizing the generalization region of $f_t$ on target domain $\mathcal{D}_t$ by using $f_s$ with $\{x^i_s, y^i_s\}^{N_s}_{i=1}$ and $\{x^i_t, y^i_t\}^{N_t}_{i=1}$, in other words, degrade the performance of $f_t$ on $\mathcal{D}_t$ while preserving its performance on $\mathcal{D}_s$ [49, 50]. Due to increasingly stringent privacy protection policies, access to the $\mathcal{D}_s$ or $\mathcal{D}_t$ database of a user is more and more difficult [28]. Thus, we introduce IP protection for source-free and data-free scenarios. The objective of ***source-free model IP protection*** is to minimize the $f_t$ generalization region of a designated target domain $\mathcal{D}_t$ by utilizing $f_s$ with $\{x_t^i\}_{i=1}^{N_t}$. ***Data-free model IP protection*** is an extreme case. The objective is to minimize the generalization bound of $f_t$ by solely utilizing $f_s$, without access to $\mathcal{D}_s$ and $\mathcal{D}_t$.

To mitigate the risk of losing valuable knowledge stored in the model parameters, we initiate our approach with unstructured pruning of the model. The lottery ticket hypothesis, proposed by [13], is widely acknowledged as a fundamental concept in the field of model pruning. Building upon this foundation, we extend our *Inverse Transfer Parameter Hypothesis* as Hypothesis 1 in alignment with the principles presented in [13].

**Hypothesis 1** (*Inverse Transfer Parameter Hypothesis*)**.** *For a dense neural network $f_s$ well-trained on the source domain $\mathcal{D}_s$, there exists a sub-network $f_{sub}$ like this: while $f_{sub}$ achieves the same test accuracy as $f_s$ on $\mathcal{D}_s$, its performance significantly degrades on the target domain $\mathcal{D}_t$. The pruned parameters of $f_s$ relative to $f_{sub}$ are crucial in determining its generalization capacity to $\mathcal{D}_t$.*

### 3.2. Source-Avaliable Model IP Protection

To verify the soundness of Hypothesis 1, we first design the source-available MAsk Pruning (SA-MAP). $f_t$ has the same architecture as $f_s$ and is initialized with a well-trained checkpoint of $\mathcal{D}_s$. To maximize the risk of the target domain and minimize it in the source domain, we prune the $f_t$'s parameters by updating a binary mask $M(\theta_M)$ of it and get a sub-network $f_{sub}$ by optimizing the objective:

$$\begin{aligned}\mathcal{L}_{SA}(f_t;\mathcal{X}_s,\mathcal{Y}_s,\mathcal{X}_t,\mathcal{Y}_t) = \frac{1}{N_s}\sum_{i=1}^{N_s} KL\left(p_t^S \| y_s\right) \\ -min\{\lambda \cdot \frac{1}{N_t}\sum_{i=1}^{N_t} KL\left(p_t^T \| y_t\right), \alpha\}\end{aligned} \tag{1}$$

where $KL(\cdot)$ presents the Kullback-Leibler divergence, $\{x_s^i, y_s^i\}_{i=1}^{N_s}$ and $\{x_t^i, y_t^i\}_{i=1}^{N_t}$ mean $N_s/N_t$ data and labels sampled from source domain $\mathcal{D}_s$ and target domain $\mathcal{D}_t$, respectively. $p_t^S = f_t(x_s)$, and $p_t^T = f_t(x_t)$. $\alpha$ and $\lambda$ are the upper bound and scaling factor, respectively, which aim to limit the over-degradation of domain-invariant knowledge. We set $\alpha = 1.0$ and $\lambda = 0.1$.

### 3.3. Source-Free Model IP Protection

Under the source-free setting, we have no access to $\{x_s^i, y_s^i\}_{i=1}^{N_s}$. To address this, we construct a replay-based source generator module to synthesize $N_s$ pseudo-source domain data $\{x_s^{i\prime}, y_s^{i\prime}\}_{i=1}^{N_s}$. As illustrated in Fig. 2, SF-MAP is deformed by removing the Diversity Generator module and employing unlabeled target data.

**Algorithm 1** SF-MAP in Source-Free Model IP Protection

```
Require: The target dataset $\mathcal{X}_t$, source network $f_s(x;\theta_s)$,
    target network $f_t(x;\theta_t)$, pre-trained model parameters $\theta_0$,
    fresh generator $G_f(z;\theta_f)$, memory generator $G_m(z;\theta_m)$,
    encoder $E_m(x;\theta_e)$, mask $M(\theta_M)$, gaussian noise $z_f$ and $z_m$.
Initialize $\theta_s$ and $\theta_t$ with $\theta_0$ and fix them
while not converged do
    Generate sample $x_f = G_f(z_f)$, $x_m = G_m(z_m)$
    Update $\theta_f$ by $x_f$ as Eq. (2)
    Concatenate synthetic data $x_s'$ by $x_f$ and $x_m$
    Update $\theta_m$ and $\theta_e$ by $x_s'$ as Eq. (3)
    Update $\theta_M$ using $x_s'$, and $x_t$ as Eq. (4)
end while
return Learned mask parameters $\theta_M$
```

Source generator module is composed of two generators to synthesize source feature samples, the fresh generator $G_f$ synthesizes samples with novel features, and the memory generator $G_m$ replays samples with origin features. Before sampling, we first train the fresh generator $G_f$ as Eq. (2). The objective of $G_f$ is to bring novel information to $f_t$. To make $G_f$ synthesize the source-style samples, we leverage the loss function of Eq. (2). The first two items of Eq. (2) derive from [4], called predictive entropy and activation loss terms. These terms are designed to encourage the generator to produce high-valued activation maps and prediction vectors with low entropy, that is, to keep the generated samples consistent with the characteristics of the origin samples. As for the third item, $JS$ denotes the Jensen-Shannon divergence, encouraging $f_t$ to obtain consistent results with $f_s$.

$$\mathcal{L}_f = \frac{1}{N}\sum_{i=1}^{N}\left[\lambda_1 t_T^i \log\left(p_s^{i\prime}\right) - \lambda_2 \mathcal{H}\left(p_s'\right) + JS\left(p_s' \| p_t'\right)\right] \tag{2}$$

where $p_s' = f_s(x_f)$, and $p_t' = f_t(x_f)$. $x_f = G_f(z_f)$ means the generated novel sample by a gaussian noise $z_f \sim \mathcal{N}(0,1)$, while $t_T^i = argmax\left(p_s^{i\prime}\right)$. $\mathcal{H}(\cdot)$ denotes the entropy of the class label distribution.

Along with the fresh generator $G_f$, we optimize the memory generator $G_m$ and encoder $E_m$ as Eq. (3), which aims to replay the features from earlier distributions. To preserve the original features, we utilize the L1 distance to measure the similarity between the generated samples and reconstructed samples. The loss is defined as follows:

$$\mathcal{L}_m = \frac{1}{N}\sum_{i=1}^{N}[||x_s' - x_{re}||_1 + \sum_{l \in L}||f_s(x_s')_l - f_s(x_{re})_l||_1] \tag{3}$$

where $||\cdot||_1$ means the L1 distance. $L$ is the selected layer set of $f_s$. $x_{re} = D_m(E_m(x_s'))$ means reconstructed sample from the encoder-decoder structure. $x_s'$ denotes the input synthetic sample concatenated by $x_f$ and $x_m$. $x_m = G_m(z_m)$ means memory sample, and $z_m \sim \mathcal{N}(0, 1)$.

After the generation process, we choose unlabeled target samples $\{x_t^i\}_i^{N_t}$ and synthetic source samples $\{x_s^{i\prime}\}_i^{N_s'}$ to train the target model $f_t$ based on Hypothesis 1, which is detailed in Algorithm 1. Adhering to the procedure in SA-MAP, we still employ the binary mask pruning strategy and update a $M(\theta_M)$ of $f_t$ as follows:

$$\begin{aligned}\mathcal{L}_{SF}(f_t, f_s; \mathcal{X}_s', \mathcal{X}_t) = \frac{1}{N_s'}\sum_{i=1}^{N_s'} KL\left(p_t'' \| \, p_s''\right) \\ -min\{\lambda \cdot \frac{1}{N_t}\sum_{i=1}^{N_t} KL\left(p_t^T \| y_{psd}\right), \beta\} \quad (4)\end{aligned}$$

$$y_{psd} = \begin{cases} f_s(x_t), & \text{if conf}(p_s^T) > \Delta \\ \frac{1}{n}\sum_i f_s(Aug_i\,(x_t)), & \text{otherwise,} \end{cases}$$

where $p_s'' = f_s(x_s')$, $p_t'' = f_t(x_s')$, $p_t^T = f_t(x_t)$, $p_s^T = f_s(x_t)$, respectively. To improve pseudo labels $y_{psd}$, we utilize a set of $i$ data augmentation $Aug_i$, when the prediction confidence conf($p_s^T$) is lower than a threshold $\Delta$. $\lambda$ and $\beta$ denote a scaling factor and an upper bound, respectively. We set $\lambda = 0.1$ and $\beta = 1.0$.

### 3.4. Data-Free Model IP Protection

When faced with challenging data-free situations, we adopt an exploratory approach to reduce the generalization region. Inspired by [52], we design a diversity generator $G_d$ with learnable mean shift $\theta_\mu$ and variance shift $\theta_\sigma$ to extend the pseudo source samples to neighborhood domains $D_{nbh}$ of variant directions. The objective is to create as many neighboring domains as feasible to cover the most target domains and limit the generalization region. After generation, we concatenate samples with distinct directions as the whole pseudo auxiliary domain.

The latent vector $z_i$ of $i$-th sample $x_i$ from the dataset $\mathcal{X}$ is generated by the feature extractor $g_s : \mathcal{X} \to \mathbb{R}^d$ of the source model $f_s = h_s(g_s(x))$, where $h_s : \mathbb{R}^d \to \mathbb{R}^k$ means the classifier, $d$ and $k$ mean the dimension of latent space and class number, respectively. This component is designed to learn and capture potential features for the generator $G_d$ in a higher-dimensional feature space. Eq. (5) applies mutual information (MI) minimization to ensure variation between produced and original samples, ensuring distinct style features.

$$\mathcal{L}_{MI} = \frac{1}{N}\sum_{i=1}^{N}[\log q\left(z_i' \mid z_i\right) - \frac{1}{N}\sum_{j=1}^{N}\log q\left(z_j' \mid z_i\right)] \quad (5)$$

However, the semantic information between the same classes should be consistent. So we enhance the semantic consistency by minimizing the class-conditional maximum mean discrepancy (MMD) [44] in the latent space to enhance the semantic relation of the origin input sample $x$ and the generated sample $x_g$ in Eq. (6), and $x_g = G_d(x)$.

$$\mathcal{L}_{sem} = \frac{1}{C}\sum_{k=1}^{C}[\| \frac{1}{N_k}\sum_{i=1}^{N_k}\phi\left(z_i^k\right) - \frac{1}{N_k'}\sum_{j=1}^{N_k'}\phi\left(z_j^{k\prime}\right)\|^2] \quad (6)$$

where $z^k$ and $z^{k\prime}$ denote the latent vector of class $k$ of $x$ and $x_g$. $q(\cdot)$ means an approximate distribution and $\phi(\cdot)$ means a kernel function. $N_k$ and $N_k'$ are the number of origin and generated samples for class $k$, respectively.

To generate diverse feature samples, we constrain different generation directions $n_{dir}$ of the gradient for the generation process detailed in supplementary material. The optimization process follows the gradient because it is the most efficient way to reach the goal. In this case, all the generated domains will follow the same gradient direction [49]. So we restrict the gradient to get neighborhood domains with diverse directions. We split the generator network $G_d$ into $n_{dir}$ parts. We limit direction $i$ by freezing the first $i$ parameters of convolutional layers. The gradient of the convolutional layer parameters is frozen during training, limiting the model's learning capabilities in that direction.

## 4. Experiments

### 4.1. Implementation Details

**Experiment Setup.** Building upon existing works, we select representative benchmarks in transfer learning—the digit benchmarks (MNIST (MN) [11], USPS (US) [20], SVHN (SN) [36], MNIST-M (MM) [15]) and CIFAR10 [22], STL10 [9] VisDA-2017 [37] for object recognition. For IP protection task, we employ the VGG11 [43], VGG13 [43], and VGG19 [43] backbones, which is the same as [50]. The ablation study additionally evaluates on ResNet50 [18], ResNet101 [18], SwinT [32] and Xception [7] backbones. We mainly compare our MAP with the NTL [49] and CUTI [50] baselines. We leverage the unitive checkpoints trained on supervised learning (SL) to initialize. To fairly compare in the source-free scenario, we replace the source and target data with synthetic samples with the generator in Section 3.3 for all baselines. Experiments are performed on Python 3.8.16, PyTorch 1.7.1, CUDA 11.0, and NVIDIA GeForce RTX 3090 GPU. For each set of trials, we set the learning rate to 1e-4, and the batch size to 32.

**Evaluation Metric.** Existing works [49, 50] leverage the *Source/Target Drop* metric ($Drop_s/Drop_t$), by quantifying the accuracy drop in the processed model compared to the original source model $f_s$ accuracy ($Acc_s/Acc_t$), to verify the effectiveness. However, these two separate metrics make it difficult to evaluate the effectiveness of the method as a whole because performance degradation on il-

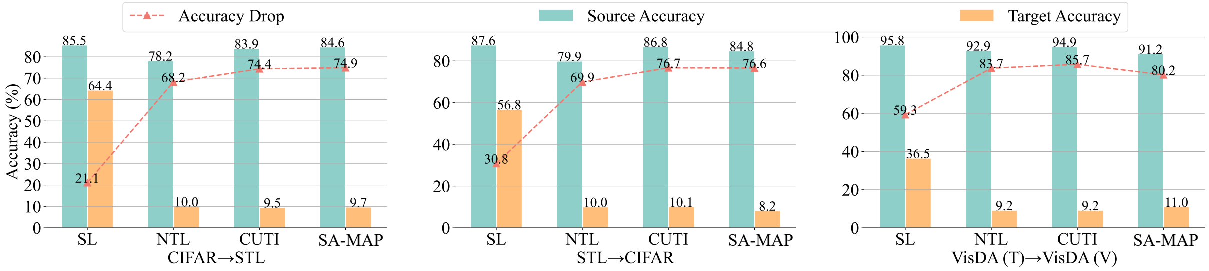

Figure 3. The accuracy of SL, NTL, CUTI, and SA-MAP on CIFAR10→STL10, and VisDA-2017 (T→V). The '→' represents the source domain transfer to the target domain. And the green bar, orange bar and red line present the accuracy of the corresponding methods in the source domain, target domain, and relative degradation (Source Accuracy - Target Accuracy), respectively.

legal target domains has the risk of destroying source domain knowledge. In order to realize the trade-off, we propose *ST-D* as Eq. (7). A lower *ST-D* denotes enhanced IP protection, with a minimal drop on the source domain and a maximum on the target.

$$\textit{ST-D} = \frac{Drop_s \ / \ Acc_s}{Drop_t \ / \ Acc_t} \tag{7}$$

## 4.2. Result of MAP in Source-Available Situation

We first conduct experiments in the source-available situation to verify the effectiveness of our Hypothesis 1. As stated before,

we introduce the SA-MAP and optimize a binary mask to realize model IP protection and compare NTL, CUTI, and SA-MAP on digit datasets. Results in Table 1 show that SA-MAP achieves better Source Drop (-0.3%) and *ST-D* (-0.004), indicating the most deterioration in the target domain and the least in the source domain. It is noteworthy that SA-MAP even outperforms the origin model in the source domain (-0.3%). We speculate this may be the result of pruning, which removes redundant parameters.

Fig. 3 assess method performance on CIFAR10, STL10, and VisDA-2017 benchmarks independently. We exploit VGG13 on CIFAR10→STL10 experiment, and VGG19 on VisDA-2017 (T→V). For a fair comparison, we utilize the same training recipe in [50].

A clear observation is that the pre-trained source model has good generalization performance on these three unauthorized tasks, seriously challenging the model IP. After performing protection, both baseline methods and our SA-MAP effectively reduce the performance on the target domain. SA-MAP achieves comparable or better results compared to baseline methods, which basically demonstrates our Hypothesis 1.

| Methods | Soure | Source Drop↓ | Target Drop↑ | *ST-D*↓ |
|---|---|---|---|---|
| NTL [49] | MT | 1.5 (1.5%) | 50.9 (77.6%) | 0.019 |
| | US | **-0.2 (-0.2%)** | **46.3 (84.0%)** | **-0.024** |
| | SN | 0.8 (0.9%) | **50.0 (85.2%)** | 0.011 |
| | MM | 2.0 (2.1%) | 59.7 (79.2%) | 0.027 |
| | Mean | 1.0 (1.1%) | **51.7 (81.5%)** | 0.013 |
| CUTI [50] | MT | 0 (0%) | **52.7 (80.0%)** | 0 |
| | US | -0.1 (-0.1%) | 42.3 (78.6%) | -0.013 |
| | SN | 0.3 (0.3%) | 48.3 (82.3%) | 0.036 |
| | MM | 0.8 (0.8%) | 60.1 (80.0%) | 0.010 |
| | Mean | 0.3 (0.3%) | 50.9 (80.2%) | 0.004 |
| SA-MAP (ours) | MT | **-0.1 (-0.1%)** | 51.0 (77.8%) | **-0.013** |
| | US | 0 (0%) | 45.2 (82.1%) | 0 |
| | SN | **-0.8 (-0.9%)** | 49.6 (84.4%) | **-0.012** |
| | MM | **-0.1 (-0.1%)** | **60.4 (80.2%)** | **-0.012** |
| | Mean | **-0.3 (-0.3%)** | 51.6 (81.1%) | **-0.004** |

Table 1. SA-MAP results in source-available situation. Note that the detailed version, likes the form of Table 2, is in the supplementary. The '↓' denotes a smaller number giving a better result, and the '↑' means the opposite. The best performances are bolded.

## 4.3. Result of MAP in Source-Free Situation

We then perform model IP protection in the challenging source-free situation and present our SF-MAP solution. We train NTL, CUTI, and SF-MAP with the same samples from Section 3.3 for fairness. Table 2 illustrates a considerable *source drop* of NTL (68.9%) and CUTI (87.4%) lead to 1.01 and 1.08 *ST-D*, respectively. While SF-MAP achieves better *Source Drop* (8.8%) and *ST-D* (0.24). The backbone in Fig. 4 is configured to correspond with the experimental setup described in Section 4.2. In particular, SF-MAP exhibits higher performance with relative degradations of 38.0%, 51.7%, and 64.9%, respectively.

We attribute impressive results to the binary mask pruning strategy. Arbitrary scaling of network parameters using continuous masks or direct adjustments has the potential to catastrophic forgetting. Precisely eliminating parameters through a binary mask offers a more effective and elegant solution for IP protection, while concurrently preventing the loss of the network's existing knowledge.

## 4.4. Result of MAP in Data-Free Situation

We next present our DF-MAP in the extremely challenging data-free setting. Section 4.3 indicates that the current techniques are not suited to source-free scenarios. Due to the absence of relevant research to our best knowledge, we refrain from specifying or constructing the baseline in the

| Methods | Source/Target | MT | US | SN | MM | Source Drop↓ | Target Drop↑ | *ST-D*↓ |
|---|---|---|---|---|---|---|---|---|
| NTL [49] | MT | 98.9 / 41.3 | 96.3 / 38.9 | 36.3 / 19.0 | 64.9 / 11.1 | 57.6 (58.2%) | 49.5 (65.1%) | 0.89 |
| | US | 90.0 / 33.2 | 99.7 / 40.0 | 32.8 / 6.8 | 42.4 / 10.8 | 59.7 (59.9%) | 38.1 (69.2%) | 0.86 |
| | SN | 68.2 / 20.5 | 75.0 / 32.7 | 92.0 / 19.4 | 32.8 / 9.2 | 72.6 (78.9%) | 37.9 (64.5%) | 1.22 |
| | MM | 97.5 / 11.3 | 88.3 / 30.7 | 40.2 / 19.0 | 96.8 / 20.6 | 76.2 (78.7%) | 55.0 (73.0%) | 1.08 |
| | Mean | / | / | / | / | 66.5 (68.9%) | 45.1 (68.0%) | 1.01 |
| CUTI [50] | MT | 98.9 / 13.0 | 96.3 / 14.1 | 36.3 / 19.0 | 64.9 / 11.2 | 85.9 (86.9%) | **51.1 (77.5%)** | 1.12 |
| | US | 90.0 / 10.7 | 99.7 / 7.8 | 32.8 / 6.6 | 42.4 / 10.6 | 91.9 (92.2%) | **53.1 (83.1%)** | 1.11 |
| | SN | 68.2 / 9.3 | 75.0 / 14.1 | 92.0 / 13.6 | 32.8 / 9.4 | 78.4 (85.2%) | **47.7 (81.4%)** | 1.05 |
| | MM | 97.5 / 11.4 | 88.3 / 14.1 | 40.2 / 19.0 | 96.8 / 14.2 | 82.6 (85.3%) | **60.5 (80.3%)** | 1.06 |
| | Mean | / | / | / | / | 84.7 (87.4%) | **53.1 (80.6%)** | 1.08 |
| SF-MAP (ours) | MT | 99.2 / 90.2 | 96.3 / 59.9 | 36.7 / 19.4 | 64.8 / 24.5 | **9.0 (9.0%)** | 31.3 (47.5%) | **0.19** |
| | US | 90.0 / 67.3 | 99.7 / 83.5 | 32.8 / 7.1 | 42.4 / 30.8 | **16.2 (16.2%)** | 20.0 (36.3%) | **0.45** |
| | SN | 68.2 / 34.8 | 75.0 / 52.5 | 91.4 / 87.2 | 32.8 / 32.0 | **4.2 (4.6%)** | 25.0 (32.2%) | **0.12** |
| | MM | 97.6 / 93.3 | 88.5 / 49.0 | 40.2 / 22.1 | 97.0 / 91.8 | **5.2 (5.4%)** | 19.6 (27.4%) | **0.20** |
| | Mean | / | / | / | / | **8.7 (8.8%)** | 24.0 (35.9%) | **0.24** |

Table 2. SF-MAP results in source-free situation. The left of '/' represents the origin source model accuracy with supervised learning, and the right of '/' denotes the accuracy of NTL, CUTI, and SF-MAP trained on the source-free setting. We synthesize samples with source domain features as pseudo-source domains and train with target data. The best performances are bolded.

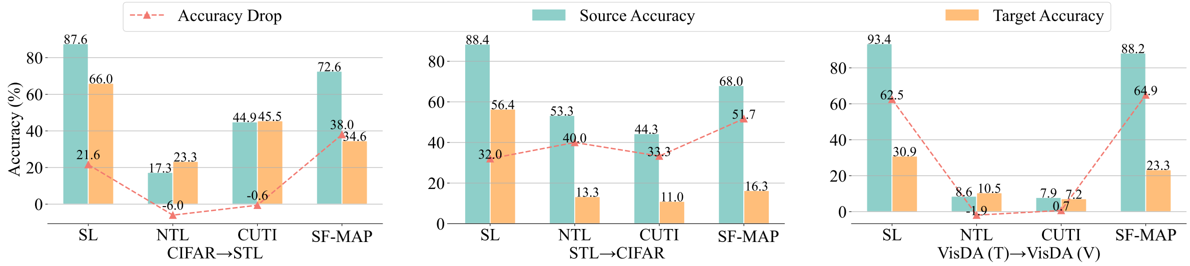

Figure 4. The accuracy of SL, NTL, CUTI, and SF-MAP on CIFAR10→STL10, and VisDA-2017 (T→V). The green bar, orange bar and red line presents the accuracy of the source domain, target domain, and relative degradation, respectively.

data-free situation. Table 3 indicates DF-MAP achieved IP protection by achieving a lower decrease in source domains and a higher drop in target domains, as illustrated by *ST-D* being less than 1.0 for all sets of experiments.

## 4.5. Result of Ownership Verification

After the above, we additionally conduct an ownership verification experiment of MAP using digit datasets and VGG11 backbone. Following existing work [49], we apply a watermark to source domain samples, treating it as an unauthorized auxiliary domain. As shown in Table 4, MAP performed 1.9% better than the second, which demonstrates the utility of this model IP protection approach.

## 4.6. Ablation Study

**Backbone.** To verify the generality of MAP for different network architectures, we examine it for several backbones, including VGG11, VGG13, VGG19 [43], ResNet18, ResNet34 [18], Swin-Transformer [32], and Xception [7]. Experiments are conducted on STL10 → CIFAR10. We evaluate model accuracy on the target domain with minimal source domain influence. Fig. 5 (a) illustrates that SF-MAP achieves consistently lower accuracy in unauthorized target domains than the origin model in supervised learning, demonstrating its universality for different backbones.

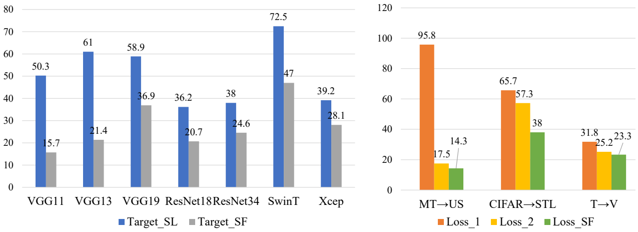

Figure 5. (a) (**left**) The accuracy (%) of origin SL and the SF-MAP model with different backbones on target of STL10 → CIFAR10 datasets. (b) (**right**) The accuracy (%) of SF-MAP with different losses on the target domain of MT → US, CIFAR10 → STL10, and VisDA-2017 (T → V).

**Loss Function.** Eq. (4) suggests $\mathcal{L}_{SF}$ shaped as $\mathcal{L}_1+\mathcal{L}_2$, where $\mathcal{L}_1 = KL\left(p_s'' \| \, p_t''\right)$ and $\mathcal{L}_2 = -KL\left(p_t^T \| y_{psd}\right)$. We conduct ablation studies to verify each loss component's contribution. We utilize $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_{SF}$ to train SF-MAP on MN→US, CIFAR10→STL10, and VisDA-2017 (T→V). According to Fig. 5 (b), the result on $\mathcal{L}_1$ has the minimum drop to the target domain due to poor simulation of its features. The result on $\mathcal{L}_{SF}$ shows the greatest target

| Methods | Source/Target | MT | US | SN | MM | Source Drop↓ | Target Drop↑ | *ST-D*↓ |
|---|---|---|---|---|---|---|---|---|
| DF-MAP (ours) | MT | 99.1 / 95.0 | 96.8 / 72.0 | 37.1 / 15.1 | 67.5 / 14.7 | 4.1 (4.1%) | 33.2 (37.8%) | 0.11 |
|  | US | 89.4 / 83.8 | 99.8 / 99.5 | 34.9 / 31.0 | 33.8 / 16.8 | 0.3 (0.3%) | 8.8 (10.3%) | 0.03 |
|  | SN | 58.6 / 46.6 | 70.9 / 59.2 | 91.5 / 76.3 | 29.5 / 24.2 | 15.2 (16.6%) | 9.7 (18.2%) | 0.91 |
|  | MM | 98.8 / 94.8 | 84.4 / 61.3 | 40.2 / 28.8 | 96.5 / 94.7 | 1.8 (1.9%) | 12.8 (17.2%) | 0.11 |
|  | Mean | / | / | / | / | 5.4 (5.7%) | 16.1 (20.9%) | 0.27 |

Table 3. DF-MAP results in the data-free situation. The right of '/' denotes the accuracy of DF-MAP is trained on the data-free setting, which cannot attain any data or labels. The '↓' means a smaller number gives a better result, and the '↑' means the opposite.

| Source | Avg Drop |  |  |  |
|---|---|---|---|---|
|  | SL | NTL | CUTI | MAP |
| MT | 0.2 | 87.9 | 87.7 | 88.6 |
| US | 0.1 | 85.7 | 93.0 | 92.5 |
| SN | -0.8 | 66.4 | 46.9 | 47.5 |
| MM | 4.4 | 83.9 | 79.6 | 79.2 |
| CIFAR | 0 | 27.4 | 38.4 | 56.2 |
| STL | -6.7 | 54.8 | 62.0 | 60.1 |
| VisDA | 0.1 | 0.1 | 22.4 | 19.1 |
| Mean | -0.3 | 58.0 | 61.4 | 63.3 |

Table 4. Ownership verification of SA-MAP. Avg Drop presents the accuracy drop between source domain and watermarked auxiliary domain. Note that the detailed version is in the supplementary.

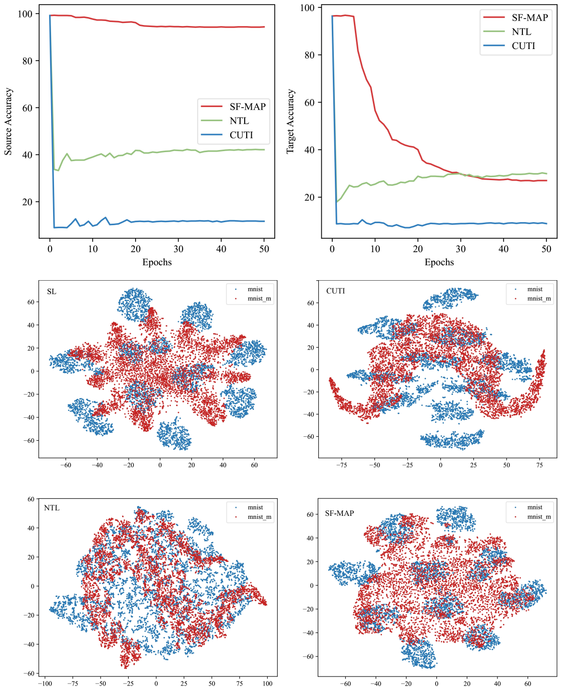

Figure 6. Visualization analysis. (a) (**top**) Converge analysis diagram. The convergence of the source accuracy and the target accuracy in the training process is exhibited. (b) (**bottom**) T-SNE visualization diagram of SL, NTL, CUTI, and SF-MAP.

drop without affecting the source domain, but $\mathcal{L}_2$ significantly degrades source performance, as detailed in the supplementary. The $\mathcal{L}_{SF}$ is more accurate than $\mathcal{L}_2$ in recognizing domain-invariant characteristics, eliminating redundant parameters, and degrading the target domain performance.

**Visualization.** Fig. 6 (a) illustrates the MN→US experiment convergence analysis diagram. With SF-MAP, source, and target domain model performance is more balanced. The fact that NTL changes all model parameters may lead to forgetting the source feature and inferior results. In the absence of the real source domain, CUTI's middle domain may aggravate forgetting origin source features since synthesized source domains may have unobserved style features. Fig. 6 (b) illustrates the T-SNE figures of MN→MM. The origin supervised learning (SL) model, NTL, CUTI, and SF-MAP results are exhibited with the source domain MNIST in blue and the target domain MNISTM in red. As illustrated in Fig. 6 (b), SF-MAP's source domain data retains better clustering information than other approaches, while the target domain is corrupted.

## 5. Conclusion

Attacks on neural networks have led to a great need for model IP protection. To address the challenge, we present **MAP**, a mask-pruning-based model IP protection method stemming from our *Inverse Transfer Parameter Hypothesis*, and its expansion forms (SA-MAP, SF-MAP, and DF-MAP) under source-available, source-free, and data-free conditions. SA-MAP updates a learnable binary mask to prune target-related parameters. Based on SA-MAP, SF-MAP uses replay-based generation to synthesize pseudo source samples. We further suggest a diversity generator in DF-MAP to construct neighborhood domains with unique styles. To trade off source and target domains' evaluation, the *ST-D* metric is proposed. Experiments conducted on digit datasets, CIFAR10, STL10, and VisDA, demonstrate that MAP significantly diminishes model generalization region in source-available, source-free, and data-free situations, while still maintaining source domain performance, ensuring the effectiveness of model IP protection.

**Acknowledgment:** This work is supported by the National Natural Science Foundation of China (No.62372329), in part by the National Key Research and Development Program of China (No.2021YFB2501104), in part by the Shanghai Rising Star Program (No.21QC1400900), in part by the Tongji-Qomolo Autonomous Driving Commercial Vehicle Joint Lab Project, and in part by the Xiaomi Young Talents Program.

## References

[1] Kuluhan Binici, Shivam Aggarwal, Nam Trung Pham, Karianto Leman, and Tulika Mitra. Robust and resource-efficient data-free knowledge distillation by generative pseudo replay. In *AAAI*, 2022. 2

[2] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022. 2

[3] Laurent Charette, Lingyang Chu, Yizhou Chen, Jian Pei, Lanjun Wang, and Yong Zhang. Cosine model watermarking against ensemble distillation. In *AAAI*, 2022. 1

[4] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *ICCV*, 2019. 4

[5] Xinyun Chen, Wenxiao Wang, Chris Bender, Yiming Ding, Ruoxi Jia, Bo Li, and Dawn Song. Refit: a unified watermark removal framework for deep learning systems with limited data. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 321–335, 2021. 2

[6] Boris Chidlovskii, Stephane Clinchant, and Gabriela Csurka. Domain adaptation in the absence of source domain data. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 451–460, 2016. 3

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 5, 7

[8] Stéphane Clinchant, Boris Chidlovskii, and Gabriela Csurka. Transductive adaptation of black box predictions. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 326–331, 2016. 3

[9] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 5

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[11] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012. 5

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[13] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 2, 3, 4

[14] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015. 2

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 5

[16] Jiyang Guan, Jian Liang, and Ran He. Are you stealing my model? sample correlation for fingerprinting deep neural networks. *NeurIPS*, 2022. 1, 2

[17] Shangwei Guo, Tianwei Zhang, Han Qiu, Yi Zeng, Tao Xiang, and Yang Liu. Fine-tuning is not enough: A simple yet effective watermark removal attack for dnn models. *arXiv preprint arXiv:2009.08697*, 2020. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5, 7

[19] Yang He and Lingao Xiao. Structured pruning for deep convolutional neural networks: A survey. *arXiv preprint arXiv:2303.00566*, 2023. 3

[20] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE TPAMI*, 16(5):550–554, 1994. 5

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 1

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[23] Peixuan Li, Pengzhou Cheng, Fangqi Li, Wei Du, Haodong Zhao, and Gongshen Liu. Plmmark: a secure and robust black-box watermarking framework for pre-trained language models. In *AAAI*, 2023. 1

[24] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020. 3

[25] Zheng Li, Chengyu Hu, Yang Zhang, and Shanqing Guo. How to prove your model belongs to you: A blind-watermark based framework to protect intellectual property of dnn. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 126–137, 2019. 2

[26] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*. PMLR, 2020. 3

[27] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE TPAMI*, 44(11):8602–8617, 2021. 3

[28] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *arXiv preprint arXiv:2303.15361*, 2023. 3

[29] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *NeurIPS*, 2021. 3

[30] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *CVPR*, 2021. 3

[31] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. Membership inference attacks by exploiting loss

trajectory. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2085–2098, 2022. 2

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 5, 7

[33] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017. 2

[34] Bo Luo, Yannan Liu, Lingxiao Wei, and Qiang Xu. Towards imperceptible and robust adversarial example attacks against neural networks. In *AAAI*, 2018. 2

[35] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, 2018. 3

[36] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 5

[37] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 5

[38] Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *CVPR*, 2022. 1, 2

[39] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. *arXiv preprint arXiv:2106.15326*, 2021. 3

[40] Sanqing Qu, Guang Chen, Jing Zhang, Zhijun Li, Wei He, and Dacheng Tao. Bmd: A general class-balanced multicentric dynamic prototype strategy for source-free domain adaptation. In *ECCV*, 2022. 3

[41] Sanqing Qu, Tianpei Zou, Florian Röhrbein, Cewu Lu, Guang Chen, Dacheng Tao, and Changjun Jiang. Upcycling models under domain and category shift. In *CVPR*, 2023. 3

[42] Sanqing Qu, Tianpei Zou, Lianghua He, Florian Röhrbein, Alois Knoll, Guang Chen, and Changjun Jiang. Lead: Learning decomposition for source-free universal domain adaptation. In *CVPR*, 2024. 3

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5, 7

[44] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *JMLR*, 2010. 5

[45] Lukas Struppek, Dominik Hintersdorf, Antonio De Almeida Correia, Antonia Adler, and Kristian Kersting. Plug & play attacks: Towards robust and flexible model inversion attacks. *arXiv preprint arXiv:2201.12179*, 2022. 2

[46] Twan van Laarhoven and Elena Marchiori. Unsupervised domain adaptation with random walks on target labelings. *arXiv preprint arXiv:1706.05335*, 2017. 3

[47] YD Vybornova and DI Ulyanov. Method for protection of deep learning models using digital watermarking. In *2022 VIII International Conference on Information Technology and Nanotechnology (ITNT)*, pages 1–5. IEEE, 2022. 1

[48] Haotian Wang, Haoang Chi, Wenjing Yang, Zhipeng Lin, Mingyang Geng, Long Lan, Jing Zhang, and Dacheng Tao. Domain specified optimization for deployment authorization. In *ICCV*, pages 5095–5105, 2023. 2

[49] Lixu Wang, Shichao Xu, Ruiqi Xu, Xiao Wang, and Qi Zhu. Non-transferable learning: A new approach for model ownership verification and applicability authorization. *arXiv preprint arXiv:2106.06916*, 2021. 2, 3, 5, 6, 7

[50] Lianyu Wang, Meng Wang, Daoqiang Zhang, and Huazhu Fu. Model barrier: A compact un-transferable isolation domain for model intellectual property protection. In *CVPR*, 2023. 2, 3, 5, 6, 7

[51] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 3

[52] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *ICCV*, 2021. 5

[53] Paul Wimmer, Jens Mehnert, and Alexandru Condurache. Interspace pruning: Using adaptive filter representations to improve training of sparse cnns. In *CVPR*, 2022. 3

[54] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, 2022. 2

[55] Chaoran Yuan, Xiaobin Liu, and Zhengyuan Zhang. The current status and progress of adversarial examples attacks. In *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*, pages 707–711. IEEE, 2021. 2

[56] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia conference on computer and communications security*, pages 159–172, 2018. 2

[57] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chee-Kong Lee, and Enhong Chen. Model inversion attacks against graph neural networks. *IEEE TKDE*, 2022. 2

[58] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *ICLR*, 2016. 1