Modeling and Simulation 建模与仿真, 2025, 14(4), 272-282
Published Online April 2025 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2025.144285

Hans汉斯

基于掩码矩阵的两阶段模型功能隐藏算法

李朋朋，黄　霖，韩彦芳

上海理工大学光电信息与计算机工程学院，上海

收稿日期：2025年3月11日；录用日期：2025年4月4日；发布日期：2025年4月14日

摘　要

随着人工智能技术的快速发展，深度神经网络模型已成为重要的数字资产，保护其版权和隐蔽传输成为关键问题。传统的模型水印技术和主动保护方案虽然在一定程度上能够防止模型被盗用，但仍然存在隐蔽性差、性能下降等问题。为此，本文提出了一种基于掩码矩阵的两阶段模型功能隐藏算法，旨在解决深度神经网络模型在公共信道中的隐蔽传输问题。该方法通过两阶段的设计，能够在解码阶段对模型做到无损恢复。算法通过生成掩码矩阵隐藏秘密任务，同时引入参数统计损失约束，最小化掩码矩阵对模型参数分布的影响，提高传输过程中的隐蔽性。实验结果表明，提出方法在不同结构模型上都有优秀的表现，模型加密前后的KL散度平均值为0.0044，秘密任务恢复后的平均准确率可以达到93.08%。所提算法为DNN模型的安全隐蔽传输提供了有效的解决方案，具有广泛的应用前景。

关键词

深度神经网络，模型功能隐藏，掩码矩阵，分布统计损失，无损恢复

Mask Matrix-Based Two-Stage Model Function Hiding Algorithm

Pengpeng Li, Lin Huang, Yanfang Han

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 11th, 2025; accepted: Apr. 4th, 2025; published: Apr. 14th, 2025

Abstract

With the rapid development of artificial intelligence technology, deep neural network models have become important digital assets, making the protection of their copyright and secure transmission critical issues. Although traditional model watermarking techniques and active protection schemes can prevent model theft, they still face problems such as poor concealment and performance

文章引用: 李朋朋, 黄霖, 韩彦芳. 基于掩码矩阵的两阶段模型功能隐藏算法[J]. 建模与仿真, 2025, 14(4): 272-282.
DOI: 10.12677/mos.2025.144285

degradation. To address this, this paper proposes a two-stage model functionality hiding algorithm based on a mask matrix, aiming to solve the problem of secure transmission of deep neural network models over public channels. The proposed method enables lossless recovery of the model during the decoding stage through a two-stage design. The algorithm generates a mask matrix to hide the secret task while introducing a parameter statistical loss constraint to minimize the impact of the mask matrix on the model's parameter distribution, thus improving the concealment during transmission. Experimental results show that the proposed method performs excellently across models with different architectures, with an average KL divergence of 0.0044 before and after model encryption, and the average accuracy of the recovered secret task reaches 93.08%. The proposed algorithm provides an effective solution for the secure and concealed transmission of DNN models, with broad application prospects.

Keywords

Deep Neural Network, Model Function Hiding, Mask Matrix, Distribution Statistical Loss, Lossless Recovery

1. 引言

近年来，人工智能技术飞速发展，深度神经网络(Deep Neural Network, DNN)模型作为其核心组成部分，在计算机视觉、自然语言处理以及多模态学习等多个领域取得了显著的成就。然而，训练高性能的DNN 模型需要大量的数据、专业的知识与昂贵的硬件设备，这也使得 DNN 模型已经成为了一项重要的数字资产，急需采取措施进行保护。模型水印技术作为一种有效的保护手段，已被广泛用于 DNN 模型的版权认证。根据模型确权时是否需要了解模型内部结构和具体参数，模型水印被划分为白盒水印和黑盒水印。白盒水印方案[1] [2]直接将模型所有者的信息嵌入到模型的冗余参数中，能够在模型被盗用时通过信息提取与比对来声明模型所有权。与此不同，黑盒水印方案[3] [4]则采用设计触发器对模型进行微调的方式嵌入水印，之后仅通过 API 接口对比输入图像的真实值和预测值异同便可进行模型的版权验证。然而，模型水印技术只能在模型被盗取后发挥作用，因此被视为一种被动的保护方式，未能从源头上根本解决问题。为此，研究者们提出了模型的主动保护方案[5] [6]，通过设计带密钥的算法，主动降低模型的性能，并对授权用户与未授权用户进行区分管理。然而，主动加密后的模型往往在任何数据集上都表现出较低的性能，这使得其在传输过程中容易被检测为异常行为，进而暴露其保护状态。因此，如何安全隐蔽地传输 DNN 模型，尤其是针对秘密任务训练的模型，依然是研究者们关注的热点问题。

最近，越来越多的研究者开始将信息隐藏技术[7]引入到 DNN 模型保护问题中，提出了“模型功能隐藏”的概念。信息隐藏技术最早是为解决多媒体信息的隐蔽传输问题而提出的，DNN 模型在某种程度上也可以被视作一种多媒体信息，因此如何为 DNN 模型找到合适的载体成为关键问题。由于 DNN 模型中通常包含大量冗余参数，因此在一个 DNN 模型中嵌入另一个 DNN 模型成为了学者们的研究重点。与传统的多媒体数据隐藏方案不同的是，DNN 模型的隐藏指的是将该模型中的任务隐藏到另一个模型中，而非该模型的全部参数。在本文中，我们定义“秘密任务”为需要隐藏的任务，“普通任务”为载体 DNN 模型的任务。通过完成隐藏过程，我们得到的模型称为隐写模型。对于未授权用户，隐写模型仅能完成普通任务；而对于授权用户，隐写模型则能够同时执行普通任务和秘密任务。Zhang 等人[8]证明了秘密

任务嵌入的可行性，他们在训练过程中将性别和种族检测任务嵌入到基于人脸的年龄预测 DNN 模型中。随后，Guo 等人[9]利用神经网络的强大拟合能力同时训练普通任务和秘密任务，其中秘密任务的输出通过伪随机矩阵进行隐藏，解码则需要密钥。在文献[10]中，普通任务和秘密任务共享模型参数，但在执行秘密任务时，模型参数会被密钥重新排列。与这些方案不同的是，Li 等人[11] [12]冻结模型中的部分滤波器保持秘密任务的性能，在此基础上构建隐写模型学习普通任务，密钥则为被冻结滤波器的位置。

现有的模型功能隐藏方案通常对载体模型进行不可逆的修改，并且可能会影响两个任务的保真度。为了解决这一问题，本文提出了一种基于掩码矩阵的两阶段模型功能隐藏算法。在我们的方法中，模型发送方需要首先训练得到一个可以执行普通任务和秘密任务的含密模型。之后，我们利用一个优化函数来生成隐藏秘密任务的掩码矩阵，使秘密任务的性能得到显著下降。为了尽可能减少掩码矩阵对模型参数分布的变化，我们引入了参数统计损失进行约束。通过实验验证，本文提出的方法能够有效地隐藏秘密任务，保证任务保真度并确保在传输过程中的隐蔽性。本文的主要贡献如下：

1) 提出了一种两阶段的模型功能隐藏算法，通过掩码矩阵实现秘密任务的隐藏。该方法保证了普通任务和秘密任务的高保真度，能够无损恢复模型，并且解密后的秘密任务能够完全恢复至原始性能。

2) 引入了参数分布统计损失，约束了隐藏秘密任务时对模型参数分布的影响，从而提高了在公共信道中传输时的隐蔽性。并且提出算法并未对模型的结构进行修改，不会增加模型传输时的额外开销。

通过在多个不同结构的模型上进行实验，验证了所提方法的通用性，展示了其广泛的应用场景。

2. 所提方法

在本部分中，我们首先对发送方提供的含密模型进行处理，在不影响其在普通任务上的性能表现的前提下，隐藏其中的秘密任务。为此，我们设计了一个包含任务损失和参数分布统计损失的隐藏秘密任务损失函数。任务损失用于确保模型在普通任务上的性能保持和在秘密任务上的性能下降，而参数分布损失则对模型参数分布的变化进行约束，从而使隐写后的模型分布尽可能接近原始含密模型的分布，这样可以提高隐写过程的隐蔽性。接下来，我们介绍了掩码矩阵，它作为方法的密钥，将分发给授权的接收方。最后，我们说明了授权用户与非授权用户在使用模型时的不同情况。图 1 给出了本文方法的总体流程。

2.1. 含密模型

在本研究中，含密模型指由模型发送方训练得到的双任务深度学习模型，其具备同时执行普通任务和秘密任务的能力。为了方法的顺利开展，我们首先需要构建含密模型，这本质上是一个多任务学习框架下的参数优化问题。传统的方法通常将各任务损失进行线性加权组合后得到总的损失函数：

$$L_t\left(\theta_0\right)=aL_1\left(\theta_0\right)+bL_2\left(\theta_0\right) \tag{1}$$

式中 a 和 b 指的是平衡两个任务损失的权重，$L_1\left(\theta_0\right)$ 和 $L_2\left(\theta_0\right)$ 指的是普通任务损失和秘密任务损失。然而，这样的静态加权策略难以通过梯度优化同时达到两个任务的最佳性能，同时训练过程中超参数 a 和 b 的敏感性显著增加了训练成本。我们借鉴文献[13]中提出的同方差不确定性(Homoscedastic Uncertainty)理论，对损失函数进行重构：

$$L_t\left(\theta_0\right)=\frac{1}{\sigma_1^2}L_1\left(\theta_0\right)+\frac{1}{\sigma_2^2}L_2\left(\theta_0\right)+\log\sigma_1+\log\sigma_2 \tag{2}$$

这允许了模型基于任务内在不确定性自主调整损失权重，即通过反向传播自动学习，减少了人工调参难度从而降低了训练成本。最后，通过梯度下降算法更新模型参数，得到含密模型：

$$\theta = \theta_0 - \omega \odot \frac{\partial L_t(\theta_0)}{\partial \theta_0} \tag{3}$$

式中 θ 为含密模型的参数，ω 代表学习率，$\odot$ 表示元素积。

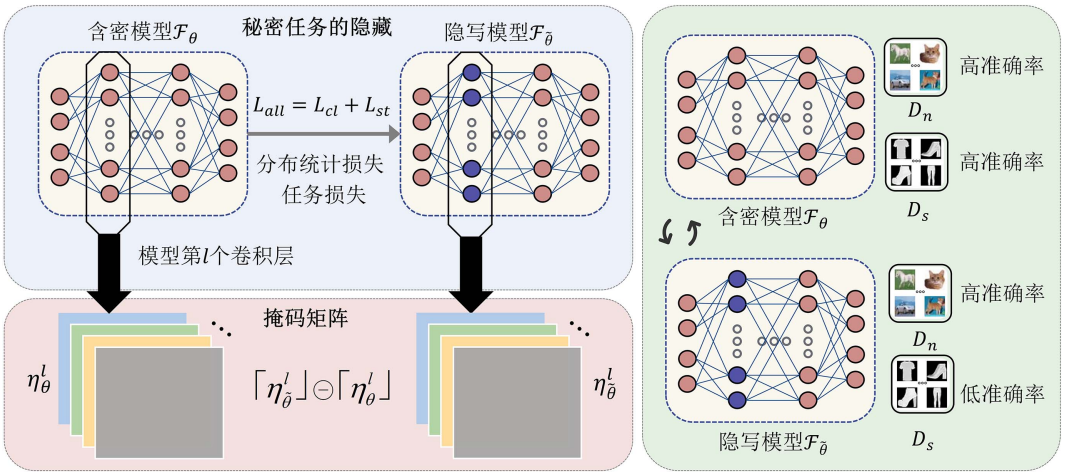


Figure 1. Overall framework diagram of the proposed method
图 1. 本文方法的流程图

2.2. 秘密任务的隐藏

为了方便描述，我们将含密模型和隐写模型分别表示为 $\mathcal{F}_\theta$ 和 $\mathcal{F}_{\tilde{\theta}}$，普通任务和秘密任务的数据集分别表示为 $D_n = \{(x_n, y_n)\}_{n=1,\cdots,\varphi_n}$ 和 $D_s = \{(x_s, y_s)\}_{s=1,\cdots,\varphi_s}$。

对于发送者提供的含密模型 $\mathcal{F}_\theta$，我们的目标是在模型对于普通任务数据集 D_n 有高性能的前提下，使其在秘密任务数据集 D_s 上的性能显著下降。我们将其描述为一个优化问题，并构建下面的优化函数，通过解决优化函数得到隐写模型 $\mathcal{F}_{\tilde{\theta}}$：

$$\mathcal{F}_{\tilde{\theta}} = \arg\min\left(\alpha\mathcal{L}_n - \beta\mathcal{L}_s + \lambda_0\left\|\mu_{\tilde{\theta}}^l - \mu_\theta^l\right\|_2 + \lambda_1\left\|\sigma_{\tilde{\theta}}^l - \sigma_\theta^l\right\|_2\right) \tag{4}$$

根据此优化函数，我们可以得到下面的损失函数：

$$L_{all} = L_{cl} + L_{st} \tag{5}$$

其中，我们引入了两种损失，L_{cl} 表示任务保真度损失，L_{st} 表示参数分布统计损失。

任务保真度损失 L_{cl} 可以使含密模型 $\mathcal{F}_\theta$ 在 D_n 上保持高性能，但在 D_s 上的性能下降，达到隐藏秘密任务的目的，通过下面的式子计算得到：

$$L_{cl} = \alpha\mathcal{L}_n - \beta\mathcal{L}_s \tag{6}$$

其中：

$$\begin{cases} \mathcal{L}_n = \dfrac{1}{\varphi_n}\displaystyle\sum_{(x_n,y_n)\in D_n} E\left(\mathcal{F}_\theta(x_n), y_n\right) \\ \mathcal{L}_s = \dfrac{1}{\varphi_s}\displaystyle\sum_{(x_s,y_s)\in D_s} E\left(\mathcal{F}_\theta(x_s), y_s\right) \end{cases} \tag{7}$$

式中 $E(\cdot,\cdot)$ 表示交叉熵损失，是一种常见的适用于分类任务的损失函数。

在采用掩码矩阵隐藏秘密任务的过程中，模型参数的修改会导致模型参数分布的变化，在传输过程中引起攻击者的注意。为了减少参数的波动幅度，我们引入了参数分布统计损失 L_{st}。该损失可以在隐藏秘密任务时对模型参数分布变化进行约束，最小化参数分布的变化，提高方法的隐蔽性，通过下面的式子计算得到：

$$L_{st}=\lambda_0\left\|\mu_{\tilde{\theta}}^{l}-\mu_{\theta}^{l}\right\|_2+\lambda_1\left\|\sigma_{\tilde{\theta}}^{l}-\sigma_{\theta}^{l}\right\|_2 \tag{8}$$

式中 $\|\cdot\|_2$ 表示 L2 范数距离，μ_{θ}^{l} 和 $\mu_{\tilde{\theta}}^{l}$ 分别代表 $\mathcal{F}_{\theta}$ 和 $\mathcal{F}_{\tilde{\theta}}$ 第 l 层参数的均值，σ_{θ}^{l} 和 $\sigma_{\tilde{\theta}}^{l}$ 分别代表 $\mathcal{F}_{\theta}$ 和 $\mathcal{F}_{\tilde{\theta}}$ 第 l 层参数的标准差，计算公式如下：

$$\begin{cases}\mu_{\theta}^{l}=\dfrac{1}{p}\sum_{i=1}^{p}\mathcal{F}_{\theta}^{l,i}\\ \mu_{\tilde{\theta}}^{l}=\dfrac{1}{p}\sum_{i=1}^{p}\mathcal{F}_{\tilde{\theta}}^{l,i}\end{cases} \tag{9}$$

$$\begin{cases}\sigma_{\theta}^{l}=\sqrt{\dfrac{1}{p}\sum_{i=1}^{p}\left(\mathcal{F}_{\theta}^{l,i}-\mu_{\theta}^{l}\right)}\\ \sigma_{\tilde{\theta}}^{l}=\sqrt{\dfrac{1}{p}\sum_{i=1}^{p}\left(\mathcal{F}_{\tilde{\theta}}^{l,i}-\mu_{\tilde{\theta}}^{l}\right)}\end{cases} \tag{10}$$

式中 $\mathcal{F}_{\theta}^{l,i}$ 和 $\mathcal{F}_{\tilde{\theta}}^{l,i}$ 分别表示 $\mathcal{F}_{\theta}$ 和 $\mathcal{F}_{\tilde{\theta}}$ 第 l 层第 i 个参数，p 表示该层参数量。

在损失函数 L_{all} 的指导下，我们可以得到隐写模型 $\mathcal{F}_{\tilde{\theta}}$，$\mathcal{F}_{\tilde{\theta}}$ 在传输时对普通任务表现出良好的性能，其隐藏的秘密任务可以被接收方通过解码手段进行获取。

2.3. 掩码矩阵

我们假设含密模型 $\mathcal{F}_{\theta}$ 共有 L 个卷积层，在进行秘密任务的隐藏时，我们只选择其中的一个卷积层进行处理，该层由模型发送者规定。这在一定程度上减少了模型的参数变化和掩码矩阵的大小，便于后续的传输和分发。我们将含密模型 $\mathcal{F}_{\theta}$ 表示为：

$$\mathcal{F}_{\theta}=\left[\eta_{\theta}^{1},\eta_{\theta}^{2},\cdots,\eta_{\theta}^{l},\cdots,\eta_{\theta}^{L}\right],\ 1\le l\le L \tag{11}$$

式中 η_{θ}^{l} 是一个向量，用来表示 $\mathcal{F}_{\theta}$ 第 l 个卷积层中所有参数的集合。由于我们只选择了模型的第 l 个卷积层进行参数更新，隐写模型 $\mathcal{F}_{\tilde{\theta}}$ 便可以表示为：

$$\mathcal{F}_{\tilde{\theta}}=\left[\eta_{\theta}^{1},\eta_{\theta}^{2},\cdots,\eta_{\tilde{\theta}}^{l},\cdots,\eta_{\theta}^{L}\right],\ 1\le l\le L \tag{12}$$

之后，我们利用下式得到掩码矩阵 M，即本文方法的密钥：

$$M=\ulcorner\eta_{\tilde{\theta}}^{l}\urcorner\ominus\ulcorner\eta_{\theta}^{l}\urcorner \tag{13}$$

式中，$\ulcorner\cdot\urcorner$ 表示矩阵转换操作，$\ominus$ 表示矩阵的逐元素差。

2.4. 秘密任务的提取

接收方在收到隐写模型 $\mathcal{F}_{\tilde{\theta}}$ 后，依据是否授权密钥(即掩码矩阵 M)，可划分为两种使用情景：经身份认证的授权用户可通过掩码矩阵 M 解密获得完整功能，而未授权用户仅能获得普通任务的性能。基于掩码矩阵 M，授权用户可以将隐写模型 $\mathcal{F}_{\tilde{\theta}}$ 无损还原成发送方传输的含密模型 $\mathcal{F}_{\theta}$，这个过程可以描述为：

$$\mathcal{F}_{\theta}=\left[\eta_{\theta}^{1},\eta_{\theta}^{2},\cdots,\eta_{\tilde{\theta}}^{l}\ominus M,\cdots,\eta_{\theta}^{L}\right],\ 1\le l\le L \tag{14}$$

由于我们在进行秘密任务的隐藏时，并没有对模型结构进行修改，所以经掩码矩阵解密后的模型在

结构上与原始含密模型 $\mathcal{F}_\theta$ 完全相同，隐藏任务的性能也恢复至原始水平。

3. 实验结果及分析

3.1. 实验设置

模型：为了验证方法的通用性，本研究选取了四个具有典型结构差异的深度神经网络模型进行实验，包括 Resnet-18 [14]，VGG-16 [15]，Densenet-121 [16]和 Mobilenet-V2 [17]。表 1 介绍了这几种深度学习模型的主要特点。

Table 1. Characteristics of the models in the experiment
表 1. 实验中用到模型的特性说明

模型	卷积层	全连接层	核心结构	参数量
Resnet-18 [14]	17	1	残差块、跳连连接	11.69 M
VGG-16 [15]	13	3	同构 3 × 3、卷积堆叠	138.0 M
Densenet-121 [16]	120	1	密集块、过渡层	7.98 M
Mobilenet-V2 [17]	52	1	倒置残差、线性瓶颈	3.50 M

数据集：我们选用 CIFAR-10 [18]作为普通任务的数据集，Fashion-MNIST [19]作为秘密任务的数据集开展实验。CIFAR-10 包含 10 类自然场景物体(飞机、汽车、鸟类等)，提供 50,000 张训练样本与 10,000 张测试样本，每幅图像为 32 × 32 像素的彩色图片。Fashion-MNIST 作为经典 MNIST 的进阶替代，由 10 类服装单品(T 恤、裤子、凉鞋等)的 70,000 张 28 × 28 灰度图构成，按 6:1 标准划分为训练/测试集。数据预处理阶段执行以下操作：1) 对 Fashion-MNIST 进行通道维度扩展(单通道到三通道)；2) 两个数据集中的图像统一上采样至 224 × 224 分辨率；3) 像素归一化。表 2 呈现了四种深度学习模型在独立训练场景下，基于这两个数据集上取得的最佳测试准确率。

参数设置：训练采用 Adam 优化器进行参数更新，学习率设置为 0.001，每次输入的 Batch 为 32。所有实验均在配备 NVIDIA GeForce RTX 2090 GPU 的计算平台完成，深度学习框架采用 Pytorch 1.8.1 版本。

Table 2. Optimal test accuracies of four models on the two datasets
表 2. 四种模型在两个数据集上的最佳测试准确率

模型	CIFAR10	Fashion-MNIST
Resnet-18 [14]	94.39%	93.69%
VGG-16 [15]	92.70%	92.25%
Densenet-121 [16]	93.43%	93.59%
Mobilenet-V2 [17]	93.29%	93.19%

3.2. 实验结果

3.2.1. 方法性能分析

根据表 3 中的实验数据分析，在使用掩码矩阵对模型参数加密后，普通任务的性能波动几乎可以忽略，其测试准确率下降区间为 0.05%~2.11% (平均降幅为 0.95%)，而秘密任务的测试准确率则显著下降

(平均降幅为 85.90%)。这表明了本文提出的方法能够在不影响模型中普通任务性能的情况下，有效地隐藏秘密任务。并且由实验结果可以看出，利用掩码矩阵还原模型参数后，普通任务和秘密任务的测试准确率均完全恢复至加密前水平，含密模型得到了无损的恢复。对比表 2 中的数据，本文提出的方法在两个任务的保真度上也有不错的表现。

Table 3. Performance analysis of the proposed method
表 3. 方法的性能分析

模型	含密模型		隐写模型		恢复模型	
	CIFAR10	Fashion-MNIST	CIFAR10	Fashion-MNIST	CIFAR10	Fashion-MNIST
Resnet-18 [14]	93.75%	93.40%	92.48%	1.53%	93.75%	93.40%
VGG-16 [15]	92.66%	92.33%	92.61%	6.53%	92.66%	92.33%
Densenet-121 [16]	93.58%	93.22%	92.46%	7.73%	93.58%	93.22%
Mobilenet-V2 [17]	93.33%	93.37%	91.22%	12.95%	93.33%	93.37%

3.2.2. 参数分布的变化

在加密模型参数从而隐藏秘密任务后，确保隐写模型在公共信道传输时不被检测到异常至关重要，这体现了提出方法对秘密任务的隐蔽传输。为此，我们采用 *KL* 散度(Kullback-Leibler divergence)作为量化指标，用于评估含密模型 $\mathcal{F}_\theta$ 和隐写模型 $\mathcal{F}_{\tilde{\theta}}$ 之间的参数分布差异。计算公式如下：

$$KL\left(\mathcal{F}_\theta^l \middle\| \mathcal{F}_{\tilde{\theta}}^l\right) = -\sum_{i=1}^{p} \mathcal{F}_\theta^{l,i} \log \mathcal{F}_{\tilde{\theta}}^{l,i} + \sum_{i=1}^{p} \mathcal{F}_{\tilde{\theta}}^{l,i} \log \mathcal{F}_\theta^{l,i} \tag{15}$$

先前的研究[20] [21]提出，训练良好的神经网络模型卷积层参数近似服从高斯分布，因此我们利用统计特征对参数分布进行拟合，并将结果代入式(15)：

$$KL\left(\mathcal{F}_\theta^l \middle\| \mathcal{F}_{\tilde{\theta}}^l\right) = \log \frac{\sigma_{\tilde{\theta}}^l}{\sigma_\theta^l} + \frac{\sigma_\theta^{l\,2} + \left(\mu_\theta^l - \mu_{\tilde{\theta}}^l\right)^2}{2\sigma_{\tilde{\theta}}^{l\,2}} - \frac{1}{2} \tag{16}$$

KL 散度越小，表明 $\mathcal{F}_\theta$ 和 $\mathcal{F}_{\tilde{\theta}}$ 的参数分布越接近，隐藏秘密任务对模型参数的修改幅度越小。表 4 展示了四种不同结构模型在隐藏秘密任务前后卷积层的 *KL* 散度平均值。实验结果表明，所有场景下的 *KL* 散度值均保持在较低水平(0.0002~0.0116)，这证实了本文提出的方法在隐藏秘密任务时对模型参数分布的影响很小。这一优势主要归功于我们引入的统计损失，它有效地约束了参数分布的变化。

Table 4. KL divergence values of different models before and after hiding the secret task
表 4. 不同模型在隐藏秘密任务前后的 *KL* 散度值

模型	Resnet-18 [14]	VGG-16 [15]	Densenet-121 [16]	Mobilenet-V2 [17]
KL 散度值	0.0056	0.0002	0.0003	0.0116

3.2.3. 鲁棒性分析

在本实验中，我们评估了本文提出的方法在三种典型攻击场景下的鲁棒性，具体包括密钥猜测攻击、模型微调攻击和模型剪枝攻击。本研究将鲁棒性定义为：在缺乏密钥的情况下，即使隐写模型遭受上述攻击后，攻击者也无法恢复秘密任务性能。

密钥猜测攻击：由于本文提出的方法具有公开性，攻击者可能通过逆向工程获取方法的具体实现细

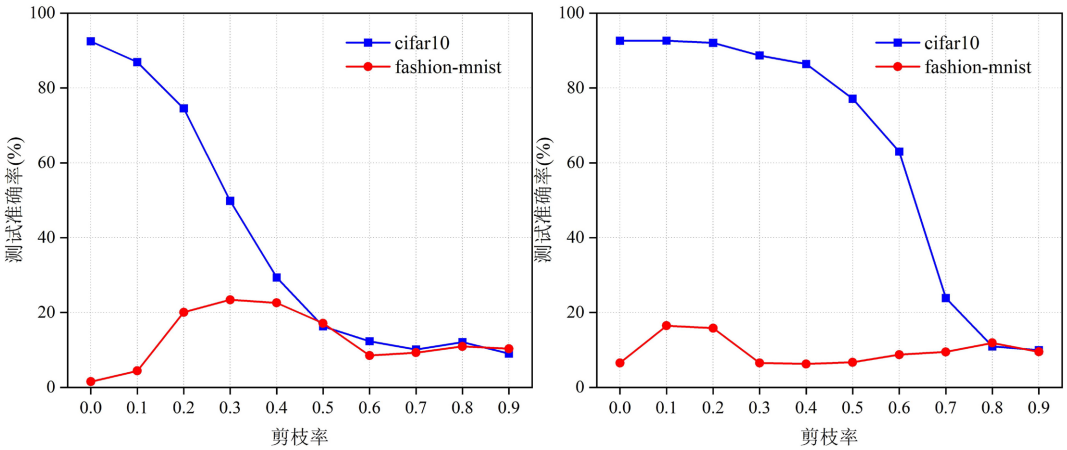
节，并尝试猜测 2.2 节中生成的掩码矩阵，恢复秘密任务的性能。为验证提出方法对该攻击的鲁棒性，我们设计如下实验：随机生成多组错误掩码矩阵，并采用本文的解密算法对隐写模型进行处理，随后在秘密任务数据集上进行性能测试。由表 5 中的实验结果可知，随机猜测的错误密钥无法有效恢复秘密任务的性能，表明了本方法对密钥猜测攻击的鲁棒性。

Table 5. Robustness of the proposed method under key guessing attack and model fine-tuning attack
表 5. 提出方法在密钥猜测攻击和模型微调攻击场景下的鲁棒性

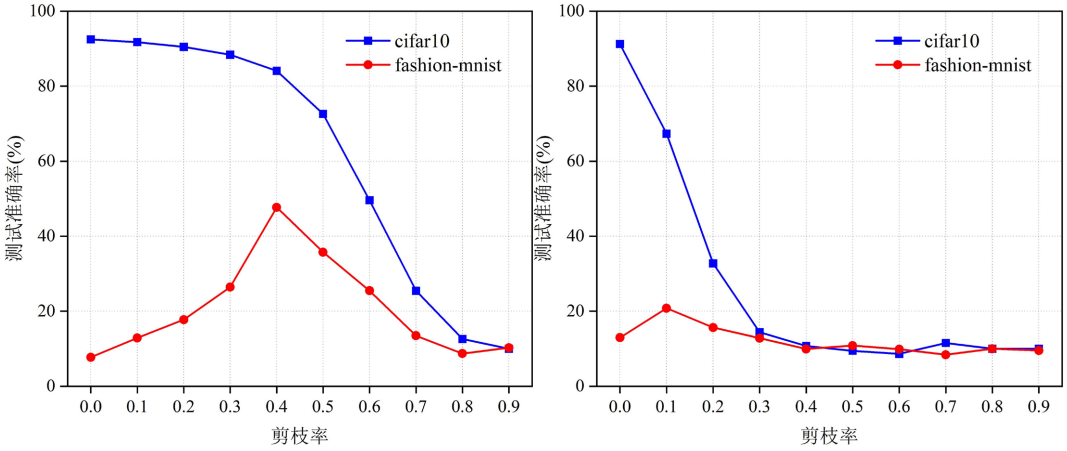
模型	数据集	攻击前	密钥猜测攻击	模型微调攻击 (10, 10%)	模型微调攻击 (20, 20%)
Resnet-18 [14]	CIFAR10	92.48%	35.92%	91.82%	91.09%
	Fashion-MNIST	1.53%	12.65%	24.27%	32.14%
VGG-16 [15]	CIFAR10	92.61%	21.37%	88.29%	89.82%
	Fashion-MNIST	6.53%	16.11%	7.57%	10.16%
Densenet-121 [16]	CIFAR10	92.46%	26.93%	93.35%	93.98%
	Fashion-MNIST	7.73%	18.55%	27.27%	40.34%
Mobilenet-V2 [17]	CIFAR10	91.22%	34.25%	92.07%	91.01%
	Fashion-MNIST	12.95%	30.74%	30.47%	45.49%

模型微调攻击：模型微调是迁移学习中的一种常见策略，通常用于在预训练模型的基础上进一步优化其性能。攻击者可能会利用模型微调作为攻击手段，对隐写模型进行微调，尝试恢复隐藏的秘密任务。在本实验中，微调的学习率设置为 0.0001，而针对微调的轮次和数据量考虑了两种情况：1) 微调轮次为 10，微调数据集为普通任务数据集的 10%；2) 微调轮次为 20，微调数据集为普通任务数据集的 20%。根据表 5 的实验结果可知，在对隐写模型进行微调后，模型在秘密任务数据集上的测试准确度无法恢复到原始水平。这表明，本文提出的方法在面对模型微调攻击时表现出较强的鲁棒性。

模型剪枝攻击：我们采用了基于 L1 范数的无结构剪枝策略来模拟模型剪枝攻击。剪枝率设置在 0.1~0.9 之间，即在实验中 10%~90%的模型参数会被置为 0，图 2 展示了在不同剪枝率下隐写模型中两个任务的性能变化。在剪枝率逐渐增大的过程中，秘密任务的测试准确率出现了小幅度上升，但远达不到加密前的数值。这表明，模型剪枝攻击无法恢复隐写模型中被隐藏的秘密任务性能，本文提出的方法在面对模型剪枝攻击时表现出较强的鲁棒性。



(a) Resnet-18 [14]　　(b) VGG-16 [15]



(c) Densenet-121 [16] (d) Mobilenet-V2 [17]

Figure 2. Robustness of the proposed method under model pruning attack
图 2. 提出方法在模型剪枝攻击场景下的鲁棒性

3.2.4. 消融实验

在本节中，我们针对 2.2 节所提出的损失函数开展了消融实验，旨在验证本文方法引入的参数分布统计损失在隐藏秘密任务中的有效性。具体而言，在模型训练过程中，我们分别采用两种不同的损失函数 $\mathcal{L}_{all}$ 和 $\mathcal{L}_{cl}$ 来得到隐写模型，并在相同的实验条件下进行对比分析，这里的实验模型为 Resnet-18。$\mathcal{L}_{all}$ 表示引入了参数分布统计损失，而 $\mathcal{L}_{cl}$ 表示仅使用了任务保真度损失。表 6 展示了在使用两种不同的损失函数后，隐写模型卷积层在加密前后的均值、标准差和 KL 散度值。实验结果表明，使用 $\mathcal{L}_{all}$ 训练得到的隐写模型的参数分布更接近含密模型，表现为参数的修改幅度更小，这充分验证了引入参数分布统计损失在增强方法隐蔽性方面的显著优势。

Table 6. Effects of statistical losses on model parameter distribution
表 6. 统计损失对模型参数分布的影响

模型	卷积层	均值	标准差	KL 散度值(×10^{-5})
含密模型 $\mathcal{F}_{\theta}$	1	0.0002	0.1299	—
	4	−0.0022	0.0515	
隐写模型 $\mathcal{F}_{\tilde{\theta}}$ （$\mathcal{L}_{all}$）	1	0.0003	0.1392	2.1124
	4	−0.0024	0.0718	1.6612
隐写模型 $\mathcal{F}_{\tilde{\theta}}$ （$\mathcal{L}_{cl}$）	1	−0.0008	0.1622	3.4995
	4	−0.0016	0.0934	3.9889

4. 总结

为了解决神经网络模型在公共信道中的隐蔽传输问题，本文提出了一种基于掩码矩阵的两阶段模型功能隐藏算法。该算法通过两阶段的处理方式，能够确保隐藏的秘密任务性能无损恢复。在掩码矩阵的生成过程中，我们设计了任务保真度损失和模型参数分布统计损失，以确保隐藏过程对模型的影响最小化。参数分布统计损失通过约束模型参数的变化，进一步增强了秘密任务在传输过程中的隐蔽性。我们在四个不同结构的神经网络模型上进行了实验，结果表明隐藏秘密任务后模型的参数分布变化几乎可以

忽略，KL 散度值平均值仅 0.0044，之后的消融实验也证明了参数分布损失的有效性。在未来的研究中，我们将扩展任务种类，进一步验证该方法在更多深度学习任务中的适用性与表现。通过进一步优化算法并应用于复杂的实际场景，我们期望能够提升方法的鲁棒性和隐蔽性，推动模型功能隐藏技术广泛应用。

基金项目

上海市教委人工智能促进科研范式改革赋能学科跃升计划项目。

参考文献

[1] Uchida, Y., Nagai, Y., Sakazawa, S. and Satoh, S. (2017) Embedding Watermarks into Deep Neural Networks. *Proceedings of the* 2017 *ACM on International Conference on Multimedia Retrieval*, Bucharest, 6-9 June 2017, 269-277. https://doi.org/10.1145/3078971.3078974

[2] Guan, X., Feng, H., Zhang, W., Zhou, H., Zhang, J. and Yu, N. (2020) Reversible Watermarking in Deep Convolutional Neural Networks for Integrity Authentication. *Proceedings of the* 28*th ACM International Conference on Multimedia*, Seattle, 12-16 October 2020, 2273-2280. https://doi.org/10.1145/3394171.3413729

[3] Adi, Y., Baum, C., Cisse, M., Pinkas, B. and Keshet, J. (2018) Turning Your Weakness into a Strength: Watermarking Deep Neural Networks by Backdooring. *Proceedings of the* 27*th USENIX Conference on Security Symposium*, Baltimore, 15-17 August 2018, 1615-1631.

[4] Jia, H., Choquette-Choo, C., Chandrasekaran, V. and Papernot, N. (2021) Entangled Watermarks as a Defense against Model Extraction. *Proceedings of the* 2021 *USENIX Conference on Security Symposium*, Online, 11-13 August 2021, 1937-1954.

[5] Xue, M., Wu, Z., Zhang, Y., Wang, J. and Liu, W. (2023) AdvParams: An Active DNN Intellectual Property Protection Technique via Adversarial Perturbation Based Parameter Encryption. *IEEE Transactions on Emerging Topics in Computing*, **11**, 664-678. https://doi.org/10.1109/tetc.2022.3231012

[6] Wu, Y., Xue, M., Gu, D., Zhang, Y. and Liu, W. (2022) Sample-Specific Backdoor Based Active Intellectual Property Protection for Deep Neural Networks. 2022 *IEEE 4th International Conference on Artificial Intelligence Circuits and Systems* (*AICAS*), Incheon, 13-15 June 2022, 316-319. https://doi.org/10.1109/aicas54282.2022.9869927

[7] Kessler, G.C. and Hosmer, C. (2011) An Overview of Steganography. *Advances in Computers*, **83**, 51-107. https://doi.org/10.1016/b978-0-12-385510-7.00002-3

[8] Cipolla, R., Gal, Y. and Kendall, A. (2018) Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 7482-7491. https://doi.org/10.1109/cvpr.2018.00781

[9] Zhang, W., Li, L. and Barni, M. (2022) Covert Task Embedding: Turning a DNN into an Insider Agent Leaking Out Private Information. *IEEE Transactions on Neural Networks and Learning Systems*, **35**, 10159-10166. https://doi.org/10.1109/TNNLS.2022.3216010

[10] Guo, Y., Qian, Z. and Zhang, X. (2022) Hiding Function with Neural Networks. 2022 *IEEE* 24*th International Workshop on Multimedia Signal Processing* (*MMSP*), Shanghai, 26-28 September 2022, 1-5. https://doi.org/10.1109/mmsp55362.2022.9949163

[11] Guo, C., Wu, R. and Weinberger, K. (2020) On Hiding Neural Networks Inside Neural Networks. arXiv: 2002.10078. https://doi.org/10.48550/arXiv.2002.10078

[12] Li, G., Li, S., Li, M., Zhang, X. and Qian, Z. (2023) Steganography of Steganographic Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**, 5178-5186. https://doi.org/10.1609/aaai.v37i4.25647

[13] Li, G., Li, S., Li, M., Qian, Z. and Zhang, X. (2023) Towards Deep Network Steganography: From Networks to Networks. arXiv: 2307.03444. https://doi.org/10.48550/arXiv.2307.03444

[14] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/cvpr.2016.90

[15] Simonyan, K. and Zisserman, A. (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. *Proceedings of the* 2015 *International Conference on Learning Representations*, San Diego, 7-9 May 2015, 1-14.

[16] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. (2017) Densely Connected Convolutional Networks. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 2261-2269. https://doi.org/10.1109/cvpr.2017.243

[17] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4510-4520. https://doi.org/10.1109/cvpr.2018.00474

[18] Krizhevsky, A. (2009) Learning Multiple Layers of Features from Tiny Images. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[19] Xiao, H., Rasul, K. and Vollgraf, R. (2017) Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv: 1708.07747. https://doi.org/10.48550/arXiv.1708.07747

[20] Huang, Z., Shao, W., Wang, X., Lin, L. and Luo, P. (2021) Rethinking the Pruning Criteria for Convolutional Neural Network. *Proceedings of the* 35*th International Conference on Neural Information Processing Systems*, Online, 6-14 December 2021, 16305-16318.

[21] Tian, J., Zhou, J. and Duan, J. (2021) Probabilistic Selective Encryption of Convolutional Neural Networks for Hierarchical Services. 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Nashville, 20-25 June 2021, 2205-2214. https://doi.org/10.1109/cvpr46437.2021.00224