Robust Ownership Verification of DNN Models Against JPEG Compression via Probability-Controlled Adversarial Attacks

Teruki Sano*†, Minoru Kuribayashi*‡, Masao Sakai*, Shuji Isobe*, Eisuke Koizumi*, Zhang Zhang*
* Tohoku University, Japan
E-mail: †sano.teruki.r2@dc.tohoku.ac.jp, ‡kminoru@tohoku.ac.jp

***Abstract*—In our previous study, a robust framework for ownership verification of deep neural network (DNN) models in image classification tasks was presented. It assumes a *gray-box* scenario in which an unauthorized user deploys a stolen model in a cloud-based setting, and outputs are returned as class probability distributions. The framework enables a legitimate owner to verify model identity without revealing the original model. In the framework, we introduced a probability-controlled *white-box* adversarial attack that crafts input images to induce a specific output probability. However, if input images are processed (e.g., via JPEG compression) before querying the model, the effects of adversarial perturbations are seriously dropped. In this paper, we propose a novel adversarial attack to support the ownership verification framework in the presence of lossy compression. Building on the iterative FGSM, we incorporate control parameters and robustness techniques to improve tolerance against JPEG compression. In addition, we study an estimation approach that adaptively controls the strength of adversarial perturbations for unknown compression parameters in the cloud-based setting. The effectiveness of the proposed adversarial attack and the adaptive method is confirmed by our intensive experiments.**

I. Introduction

Deep neural networks (DNNs) have become integral to a wide range of modern AI applications, such as image classification and speech recognition. To reduce the high cost of training such models, many providers now offer access to pre-trained models through cloud-based services known as Machine Learning as a Service (MLaaS). While this improves accessibility, it also increases the risk of unauthorized use of proprietary models, which often require substantial computational and financial resources for development.

To protect DNN model ownership, two main approaches have been widely investigated: DNN watermarking and DNN fingerprinting [1], [2]. DNN watermarking typically embeds ownership information into the model during training, for example by modifying weight parameters, inserting special neurons, or training with additional trigger samples that elicit predefined outputs. In contrast, DNN fingerprinting does not modify the model itself, but instead extracts behavioral characteristics such as responses to carefully designed queries or the geometry of decision boundaries. Both approaches, however, suffer from practical limitations. Watermarks can be weakened or erased through fine-tuning, pruning, or model compression, while fingerprinting often depends on a fixed and limited set of query samples that produce statistically unusual responses. Such atypical behaviors may be detected by unauthorized users monitoring the queries and outputs, who could then apply retraining or other transformations to evade verification [3], [4]. This limitation becomes even more critical in MLaaS environments, where service providers or adversarial users may actively monitor both inputs and outputs of the deployed model. In such cases, verification queries that deviate from normal user behavior are at high risk of being flagged or filtered, further undermining the reliability of watermarking and fingerprinting schemes.

To address these limitations, we proposed in [5] a novel ownership verification framework based on white-box adversarial attacks [6]. In the framework, a legitimate owner of a model, with full access to its internal weight parameters, generates adversarial samples that manipulate the output probability of a target class to a specified value. A key component of the method was the I-FDGSM (Iterative-Fast Dual Gradient Sign Method), which simultaneously adjusts the probabilities of both the original and target classes to ensure stealthiness. The framework allows for ownership verification in a gray-box setting without revealing the model or relying on static triggers.

In practical deployments, images are frequently subjected to lossy compression, especially JPEG, during transmission or processing. Cloud-based environments may introduce additional compression either automatically or as a defense against adversarial attacks. These real-world transformations pose a serious threat to the reliability of adversarial-sample-based verification. Previous studies have shown that adversarial perturbations are especially fragile under denoising operations such as JPEG compression, with even mild compression often neutralizing their effect [7]. This fragility limits the practical applicability of the our previous method in real-world verification pipelines involving image compression.

In this paper, we propose a novel ownership verification method that is robust against real-world image processing conditions. Specifically, we extend the I-FDGSM to a JPEG-aware variant called I-FDGSM-AJ (Against JPEG compression), which incorporates JPEG compression simulation directly into the optimization process. This design generates

979-8-3315-7206-8/25/$31.00

adversarial samples that preserve the output behavior under lossy compression. Furthermore, we propose an adaptive algorithm that adjusts the strength of the adversarial perturbations based on observations of output behavior for some queries, allowing reliable verification even when the actual compression parameters are unknown.

II. Related Work

A. Threat Model

We consider a gray-box verification setting in which the legitimate owner of a deep neural network (DNN) model attempts to verify whether a remotely deployed model is an unauthorized copy of their original model. The suspect model is assumed to be deployed as a machine learning service (e.g., MLaaS), and accessible only through an inference API that returns class probability distributions for input images.

The verification process involves three parties: (i) the **owner**, who holds the original model and can generate adversarial samples using full white-box access; (ii) the **unauthorized user**, who may have copied the model and deployed it as a cloud service; (iii) a **third party**, who acts as an external verifier and determines the legitimacy of model ownership by asking two parties for proof.

It is assumed that the unauthorized user denies access to the model's internal structure and parameters and can observe all incoming queries and returned outputs.

B. DNN Watermarking

DNN watermarking embeds identifying information into models to deter unauthorized use [1]. Techniques include black-box (input-output behavior), white-box (model parameters), and gray-box (probability outputs) approaches [8]. Backdoor-based methods, such as Adi et al. [9], insert triggers that elicit specific responses. However, several limitations exist: (i) Ownership is unverifiable if the model is published without a watermark. (ii) Since watermarks encode task-irrelevant data, they may introduce detectable patterns vulnerable to statistical analysis or retraining. (iii) Watermarks can be weakened or erased through fine-tuning [3], [4]. Additionally, embedding unrelated information may degrade the model performance in terms of accuracy.

C. DNN Fingerprinting

DNN fingerprinting aims to characterize a model by exploiting distinctive features such as decision boundaries or adversarial response patterns [10]. Unlike DNN watermarking, fingerprinting does not require modifying the model parameters, but instead relies on query–response behaviors for verification. As the query, trigger inputs are prepared in advance to assure the validity of verification. While this approach avoids embedding external information, it introduces its own challenges: if the verification queries are statistically distinguishable, unauthorized users may detect ongoing verification attempts and apply fine-tuning or retraining to alter the decision boundaries. As a result, fingerprinting methods remain vulnerable to removal or evasion, particularly when applied in realistic MLaaS environments where the adversary can monitor input–output interactions.

D. Adversarial-Sample-based Method

Sano et al. [5] proposed a framework for verifying the ownership of deep neural network (DNN) models under gray-box conditions, where only output probabilities of a suspected model can be observed via an inference API. As the full access to the model enables an owner to craft adversarial samples under the constraints of accurate probability control of specified classes, the owner can convince a third party that he/she owns the original model.

Let M be the original model, and M^{copy} a potentially unauthorized copy deployed in a cloud service. Both models perform k-class image classification, but only M^{copy} is accessible via API calls. Given an input image $\boldsymbol{x}$, the correct class is $c = \arg\max_i M(\boldsymbol{x})_i$. The owner generates an adversarial sample $\boldsymbol{x}^{\text{adv}}$ such that $M(\boldsymbol{x}^{\text{adv}}) = \tilde{\boldsymbol{p}}$ satisfies $\tilde{p}_{c'} \approx p_{c'}^{\text{target}}$ for a designated target class c', while maintaining $\arg\max_i \tilde{p}_i = c$. This constraint on the top prediction plays a critical role: it ensures that the adversarial sample remains visually and semantically similar to the original input, thereby reducing the risk of detection or rejection by unauthorized users. If the correct class remains the top-1 prediction, the sample appears normal, preventing suspicion during model queries. If the same adversarial input yields a similar probability vector under M^{copy}, the owner can confidently assert that $M^{\text{copy}} = M$.

To generate such finely tuned samples, we proposed the I-FDGSM, a white-box attack that balances the gradients of both the original and target classes using tunable coefficients. This enables accurate probability control while preserving the dominance of the original class, making the behavior difficult to replicate without access to M—thus serving as cryptographic-style proof of ownership. The I-FDGSM update rule is:

$$\begin{aligned} \boldsymbol{x}_0^{\text{adv}} &= \boldsymbol{x}, \\ \boldsymbol{x}_{N+1}^{\text{adv}} &= \text{Clip}_{\boldsymbol{x},\varepsilon}\Big\{\boldsymbol{x}_N^{\text{adv}} - \\ \alpha^{\text{com}}\ &\text{sign}\left(\beta^c \nabla_x C(\boldsymbol{x}_N^{\text{adv}}, c) + \beta^{c'} \nabla_x C(\boldsymbol{x}_N^{\text{adv}}, c')\right)\Big\} \end{aligned} \tag{1}$$

Here, $C(\boldsymbol{x}, c)$ denotes the loss function with respect to class c (e.g., cross-entropy loss), and N represents the iteration index of the update. ε is the perturbation bound, α^{com} the step size, and β^c, $\beta^{c'}$ the gradient weights for classes c and c'. Proper tuning of these parameters allows flexible control over output probabilities while minimizing perturbation and maintaining stealth.

However, this method assumes the crafted adversarial samples are plain(uncompressed) images and may not be robust to real-world transformations such as JPEG compression—a limitation addressed in this extended work.

III. Proposed Method

Unlike the previous work [5], this study considers realistic environments where adversarial samples may undergo lossy

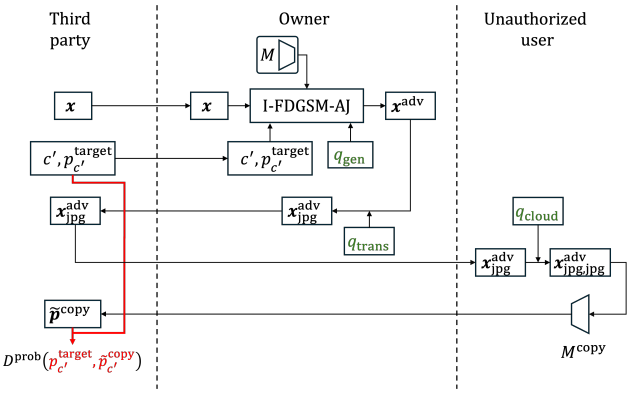


Fig. 1. Overview of the proposed verification framework under JPEG compression.

image processing such as JPEG compression. In practice, compression must be applied by different parties for distinct reasons: **(i) Owner-side compression:** The legitimate owner applies JPEG compression to reduce transmission cost. Users interacting with the deployed cloud model must comply with designated input constraints, such as accepted formats. **(ii) Unauthorized user-side compression:** An unauthorized user who controls the cloud model applies JPEG compression to remove redundancy involved in incoming queries. It is practical to reduce the influence of noise signals in the query. Since the internal processing of cloud services is typically undisclosed, the specific compression settings (e.g., quality factor (QF)) are often unknown, making robust verification even more challenging.

Adversarial samples are known to be vulnerable to lossy compression [7], as the effects of adversarial perturbations are reduced. This poses a serious challenge for the case of ownership verification in the previous method, where accurate control of the output probabilities is required. To address this challenge, we propose a robust verification framework that introduces resilience to JPEG compression. The proposed method comprises two main components:

- **I-FDGSM-AJ:** A JPEG-aware extension of I-FDGSM that incorporates simulated JPEG compression into the loop generating adversarial perturbations, allowing the generation of adversarial samples that keep the desired output behavior even after compression.
- **Adaptive Adjustment Algorithm:** When the QF of JPEG compression is unknown, this algorithm estimates an appropriate generation QF q_{gen} to ensure the generated sample remains effective even after undergoing JPEG compression.

This framework enables reliable verification of model identity under realistic conditions, including double compression, unknown QF, and intentional interference by unauthorized users, without requiring access to the internal structure of the suspected model.

We define the JPEG QF parameters as follows: q_{gen}: QF used during adversarial sample generation (by the owner), q_{trans}: QF applied during image transmission (by the owner), q_{cloud}: QF possibly applied by the cloud service upon receiving the image.

To evaluate the deviation from the target probability value, we use the following metric:

$$D^{\text{prob}}(p_{c'}^{\text{target}}, \tilde{p}_{c'}^{\text{copy}}) = \frac{p_{c'}^{\text{target}} - \tilde{p}_{c'}^{\text{copy}}}{p_{c'}^{\text{target}}}.$$

Here, $p_{c'}^{\text{target}}$ is the specified target probability for class c', and $\tilde{p}_{c'}^{\text{copy}}$ is the actual output probability from the suspected model after compression. A smaller D^{prob} indicates successful preservation of the desired probability, thereby supporting verification.

A. I-FDGSM-AJ (Against JPEG Process)

To address the vulnerability of I-FDGSM under lossy transformations, we propose *I-FDGSM-AJ*, a robust variant that incorporates JPEG compression directly into the optimization process. This method is based on the original I-FDGSM and retains the same update rule (Eq. 1) for generating perturbations. However, I-FDGSM-AJ introduces a novel parameter adjustment mechanism specifically designed for robustness against JPEG compression. As outlined in Algorithm 1, JPEG compression is inserted into each iteration of the optimization loop to simulate real-world degradation. The compression is applied using a specified JPEG QF, q_{gen}, which serves as a robustness parameter during adversarial sample generation. A critical insight in I-FDGSM-AJ is that if the actual QF applied after generation (e.g., during transmission or in the cloud) is significantly higher than q_{gen}, the perturbation may become over-amplified. This can lead to a much higher output probability than the intended value $p_{c'}^{\text{target}}$, compromising verification accuracy. Therefore, it is important that q_{gen} closely approximates the actual QF to ensure the adversarial sample maintains the desired probability behavior.

During optimization, the parameters $\alpha^{\text{com}}, \beta^{c}$, and $\beta^{c'}$ are dynamically adjusted based on the discrepancy between the compressed model output $\tilde{p}_{c'}$ and the target probability $p_{c'}^{\text{target}}$. Once the shared step size α^{com} falls below a minimal threshold (e.g., 10^{-10}), it indicates that additional perturbations have negligible effect, and the iteration is terminated.

B. Verification Process under JPEG Compression

The framework consists of two phases: (1) *Adjustment Phase*, which adaptively estimates the appropriate JPEG QF for robust adversarial sample generation, and (2) *Verification Phase*, which tests model identity based on output probabilities. The overall process is illustrated in Fig. 1.

1) Adjustment Phase for Unknown JPEG Compression: In practice, the JPEG QF used during transmission (q_{trans}) is typically known to the legitimate owner. If no further compression is applied in the cloud, the generation QF can match the transmission setting: $q_{\text{gen}} = q_{\text{trans}}$. However, cloud environments may apply additional JPEG compression with an unknown QF q_{cloud}. In such a case, even with a known

Algorithm 1 I-FDGSM-AJ

Require: Original Image: $\boldsymbol{x}$, Model: M,
Target Classes: c, c', Target Probability Value: $p_{c'}^{\text{target}}$,
Factors of I-FDGSM: $\alpha^{\text{com}}, \beta^{c} = 1, \beta^{c'} = 1$,
Averaging Interval: l, Tolerance for Error: T^{diff}
Ensure: Adversarial Image: $\boldsymbol{x}^{\text{adv}}$

$\boldsymbol{x}_0^{\text{adv}} = \boldsymbol{x}$
for $N \in \{1, 2, \cdots, N^{\max}\}$ **do**
  $\boldsymbol{x}_N^{\text{adv}} = Adv(\boldsymbol{x}_{N-1}^{\text{adv}}, M, \alpha^{\text{com}}, \beta^{c}, \beta^{c'})$
  $\boldsymbol{x}_{N,jpg}^{\text{adv}} = JPEG(\boldsymbol{x}_N^{\text{adv}}, QF = q_{gen})$
  $\tilde{\boldsymbol{p}}_{\boldsymbol{N}} = M(\boldsymbol{x}_{N,jpg}^{\text{adv}})$
  if $N \mod l = 0$ **then**
    $\tilde{\boldsymbol{p}}^{\text{mean}} = \frac{1}{l}\sum_{i=N-l+1}^{N} \tilde{\boldsymbol{p}}_{\boldsymbol{i}}$
    if $\tilde{p}_{c'}^{\text{mean}} < (1 - T^{\text{diff}}) p_{c'}^{\text{target}}$ **then**
      $\beta^{c'} \leftarrow \beta^{c'} + 1$
    else
      $\beta^{c} \leftarrow \beta^{c} + 1$
    end if
    if $(1 - T^{\text{diff}}) p_{c'}^{\text{target}} < \tilde{p}_{c'}^{\text{mean}} < (1 + T^{\text{diff}}) p_{c'}^{\text{target}}$ **then**
      $\alpha^{\text{com}} \leftarrow 0.5\ \alpha^{\text{com}}$
    end if
    if $\alpha^{\text{com}} < 10^{-10}$ **then**
      $\boldsymbol{x}^{\text{adv}} = \boldsymbol{x}_N^{\text{adv}}$
      break
    end if
  end if
end for

q_{trans}, determining an appropriate q_{gen} becomes non-trivial. The final output probability $\tilde{p}_{c'}^{\text{copy}}$ depends on the cumulative degradation introduced by both q_{trans} and q_{cloud}, relative to the simulated compression at q_{gen}. If q_{gen} is lower than the effective compression level, the perturbation may be over-preserved, resulting in a higher-than-intended $\tilde{p}_{c'}^{\text{copy}}$. Conversely, if q_{gen} is higher, the perturbation may be excessively suppressed, leading to a lower-than-target probability. To address this, the framework performs an adaptive adjustment: for each candidate q_{gen}, an adversarial sample $\boldsymbol{x}^{\text{adv}}$ is generated using I-FDGSM-AJ, where JPEG simulation with the given q_{gen} is applied during optimization. The generated sample is then JPEG compressed using q_{trans} and queried through M^{copy} to obtain the output probability $\tilde{p}_{c'}^{\text{copy}}$. The deviation from the target is computed as $D^{\text{prob}}(p_{c'}^{\text{target}}, \tilde{p}_{c'}^{\text{copy}})$. This process is repeated while refining q_{gen} until the observed output closely matches the target probability, enabling robust verification despite unknown JPEG settings.

This adaptive adjustment process is formalized in Algorithm 2. Starting from an initial q_{gen}, the framework iteratively generates adversarial samples using I-FDGSM-AJ, queries M^{copy} to obtain output probabilities, and evaluates the deviation from the target using the distance metric D^{prob}. The direction of the update is determined by the sign of the average deviation $\overline{D}$ across the image set: if $\overline{D}^{\text{prob}} > 0$, q_{gen} is decreased to amplify the perturbation effect; otherwise, q_{gen} is increased to suppress it. This rule helps guide the search toward the desired output probability. The update is repeated using a fixed step size Δq until convergence or until a maximum number of iterations is reached.

Algorithm 2 Adaptive Estimation of JPEG Quality Factor

Require: Image set $\{\boldsymbol{x}_i\}_{i=1}^{N}$, models M, M^{copy}, target probability $p_{c'}^{\text{target}}$, step size Δq
Ensure: Estimated QF q_{gen}

Initialize q_{gen}
for $t = 1$ to $T^{\max}$ **do**
  for $i = 1$ to N **do**
    $\boldsymbol{x}_i^{\text{adv}} = \text{I-FDGSM-AJ}(\boldsymbol{x}_i, M, q_{\text{gen}})$
    $D_i = D^{\text{prob}}(p_{c'}^{\text{target}}, M^{\text{copy}}(\boldsymbol{x}_i^{\text{adv}})_{c'})$
  end for
  $\overline{D} = \frac{1}{N}\sum_i D_i$
  if $\overline{D} > 0$ **then**
    $q_{\text{gen}} \leftarrow q_{\text{gen}} - \Delta q$
  else
    $q_{\text{gen}} \leftarrow q_{\text{gen}} + \Delta q$
  end if
end for

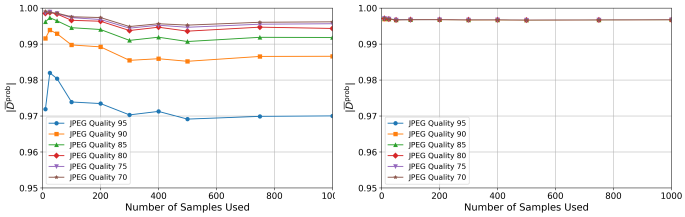
2) Verification Phase under Double JPEG Compression: Once q_{gen} is determined, the owner generates an adversarial sample $\boldsymbol{x}^{\text{adv}}$ using I-FDGSM-AJ with virtual JPEG simulation at q_{gen}. The sample is then compressed with q_{trans} and submitted to the cloud model, which may further compress it with an unknown q_{cloud}. Thus, double compressed image is input to M^{copy}, which returns a probability vector $\tilde{\boldsymbol{p}}^{\text{copy}}$. If the observed probability $\tilde{p}_{c'}^{\text{copy}}$ remains sufficiently close to the target $p_{c'}^{\text{target}}$—i.e., if D^{prob} is small—then the third party can conclude that $M^{\text{copy}} = M$. Conversely, if D^{prob} is far from zero, the probability could not be accurately controlled, indicating that the queried model differs from the owner's and verification fails.

IV. Computer Simulation

A. Performance Degradation of I-FDGSM under Compression

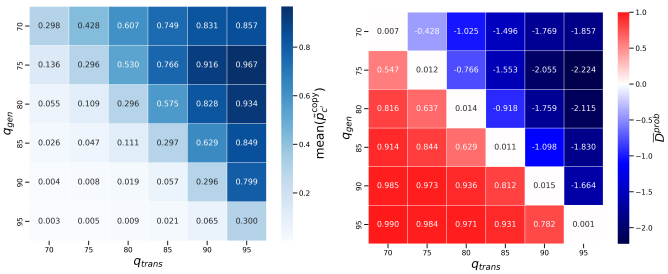
To evaluate I-FDGSM's robustness under JPEG compression, we conduct a quantitative analysis of its probability manipulation performance under varying compression conditions. In this experiment, adversarial samples were generated from the ImageNet validation set using model M (ResNet50-v1), targeting class c' with $p_{c'}^{\text{target}} = 0.3$. JPEG compression was then applied to the generated samples with QF ranging from 95 to 70. We throw some queries to M^{copy} by increasing the number of adversarial samples to observe the convergence behavior of the performance metric (averaged $|\overline{D}^{\text{prob}}|$). Note that $|\overline{D}^{\text{prob}}|$ is close to 0 if an adversarial sample is uncompressed.

The results are shown in Fig. 2. Two evaluation models were considered: the same model as the generator ($M = M^{\text{copy}} =$ ResNet50-v1), and a different model ($M^{\text{copy}} =$ ResNet50-v2), to assess robustness and transferability, respectively. Note that ResNet50-v1 and ResNet50-v2 in PyTorch share the same network architecture, but are trained with different parameter



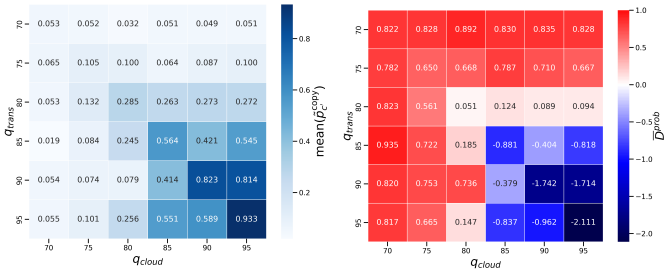
(a) same model (b) different model

Fig. 2. Variations of $|\overline{D}^{\text{prob}}|$ with respect to the number of used samples.



(a) $\text{mean}(\tilde{p}_{c'}^{\text{copy}})$ (b) $\overline{D}^{\text{prob}}$

Fig. 3. Heatmap of $\text{mean}(\tilde{p}_{c'}^{\text{copy}})$ and $\overline{D}^{\text{prob}}$ values for multiple q_{gen} and q_{trans}.



(a) $\text{mean}(\tilde{p}_{c'}^{\text{copy}})$ (b) $\overline{D}^{\text{prob}}$

Fig. 4. Heatmap of $\text{mean}(\tilde{p}_{c'}^{\text{copy}})$ and $\overline{D}^{\text{prob}}$ values for multiple q_{trans} and q_{cloud}. $q_{\text{gen}} = 80$.

initializations and recipes, resulting in distinct weight sets. In the same-model case, the absolute value $|\overline{D}^{\text{prob}}|$ converges to approximately 0.97, while in the different-model case, it remains consistently high in the range of 0.99 to 1.00. Moreover, it is observed that the value of $|\overline{D}^{\text{prob}}|$ increases as QF decreases, indicating a degradation in attack performance. These results demonstrate that the probability manipulation achieved by I-FDGSM becomes fragile under JPEG compression, especially when QF drops below 85. As a result, verification accuracy deteriorates, and the ability to distinguish between M and M^{copy} diminishes. This highlights the need for a more compression-resilient adversarial attack.

B. Verification with Single JPEG Compression

In this experiment, we evaluate the effectiveness of the proposed I-FDGSM-AJ method in a controlled setting where JPEG compression is applied only once. Specifically, we consider a scenario involving a single round of JPEG compression with no additional processing (i.e., q_{cloud} is not applied). This allows us to isolate the effect of the transmission-stage QF q_{trans} and assess how well adversarial perturbations are preserved under these conditions. Adversarial samples were generated using ResNet50-v1 via the I-FDGSM-AJ, such that the target class probability was set to $p_{c'}^{\text{target}} = 0.3$. For a given generation QF q_{gen}, adversarial samples were generated for 100 images from the ImageNet validation set. Each sample was subsequently compressed using a different QF q_{trans}, and the resulting images were input into the same model ($M = M^{\text{copy}} = \text{ResNet50-v1}$). For each combination of q_{gen} and q_{trans}, we recorded the average probability distance $\overline{D}^{\text{prob}}$ and the mean target class probability $\text{mean}(\tilde{p}_{c'}^{\text{copy}})$. These metrics were used to evaluate how well the adversarial perturbations survived the lossy compression process and whether the ownership verification conditions were still satisfied.

The results are shown in Fig. 3. When $q_{\text{gen}} = q_{\text{trans}}$, the probability is $\text{mean}(\tilde{p}_{c'}^{\text{copy}}) \approx 0.3$, and $|\overline{D}^{\text{prob}}| \approx 0$. When $q_{\text{gen}} < q_{\text{trans}}$, the probability increases ($\text{mean}(\tilde{p}_{c'}^{\text{copy}}) > 0.3$) and $\overline{D}^{\text{prob}} \ll 0$, indicating that the generated perturbations are preserved more strongly than expected. Conversely, when $q_{\text{gen}} > q_{\text{trans}}$, the probability decreases ($\text{mean}(\tilde{p}_{c'}^{\text{copy}}) < 0.3$) and $\overline{D}^{\text{prob}} \gg 0$, suggesting that the perturbation effect is excessively suppressed due to stronger-than-anticipated compression. These results demonstrate that, if q_{trans} used in the compression process is known in advance, setting $q_{\text{gen}} = q_{\text{trans}}$ can notably improve verification accuracy.

C. Verification with Double JPEG Compression

We conduct an evaluation assuming a scenario in which the owner of M^{copy} applies JPEG compression with q_{cloud} to the received adversarial samples. The experimental conditions are the same as in the previous section, except that the generation QF is fixed at $q_{\text{gen}} = 80$, and an additional JPEG compression using q_{cloud} is applied after transmission. We investigate the impact on verification performance when both q_{trans} and q_{cloud} vary.

The results are shown in Fig. 4. When both q_{trans} and q_{cloud} are greater than q_{gen}, and at least one of them is close to q_{gen}, the perturbation effect is preserved, yielding $\overline{D}^{\text{prob}} \approx 0$ and enabling accurate verification. This indicates that the model output under double JPEG compression is predominantly influenced by the lower of the two QFs. However, our results show that even when q_{trans} and q_{cloud} are known, accurately selecting q_{gen} to minimize $\overline{D}^{\text{prob}}$ is not straightforward. This is because the optimal q_{gen} does not always lie exactly at either value, and its impact depends on the complex interaction between the two compression stages.

D. Verification under Unknown Cloud Compression

Building on the proposed adaptive algorithm, Algorithm 2, we evaluate its effectiveness in scenarios where the cloud-side QF q_{cloud} is unknown, while the transmission-side QF q_{trans} is

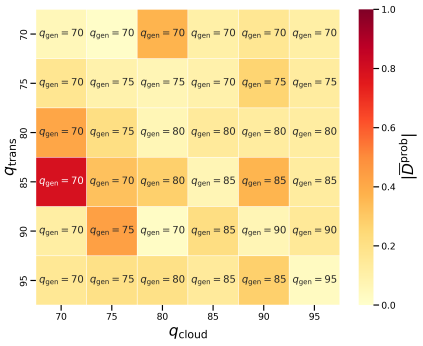


Fig. 5. Heatmap of estimated q_{gen} and the corresponding $\overline{D}^{\text{prob}}$ values for multiple q_{trans} and q_{cloud}.

known. The objective is to determine whether the generation QF q_{gen} can be adjusted using only feedback from the output probability deviation D^{prob}, without requiring knowledge of q_{cloud}. In the simulation, candidate values of q_{gen} are iteratively refined with a fixed step size $\Delta q = 5$, balancing estimation accuracy and computational efficiency. In our simulation, the maximum iteration count was set to $T^{\max} = 10$, which provided a good balance between efficiency and convergence.

The results are shown in Fig. 5. In the heatmap, the vertical and horizontal axes represent q_{trans} and q_{cloud}, respectively. Each cell displays the estimated q_{gen} obtained by the proposed algorithm, while the color indicates the corresponding value of $\overline{D}^{\text{prob}}$. The maximum observed value of $\overline{D}^{\text{prob}}$ is 0.79, occurring at the combination $(q_{\text{trans}}, q_{\text{cloud}}) = (85, 70)$. In this case, the lower bound for q_{gen} was set to 70, suggesting that further reduction in $\overline{D}^{\text{prob}}$ may be possible if smaller q_{gen} values are permitted. For all other combinations, $\overline{D}^{\text{prob}}$ remains below 0.5, indicating that the proposed algorithm effectively estimates a suitable q_{gen} and allows a successful verification. Furthermore, when $M = M^{\text{copy}}$, reducing the step size Δq can further decrease $\overline{D}^{\text{prob}}$.

In contrast, when $M \neq M^{\text{copy}}$, it is observed that $\overline{D}^{\text{prob}}$ remains close to 1 under all conditions, suggesting that accurate estimation of q_{gen} is not achievable in such cases. These results suggest that the proposed approach exhibits strong robustness with extremely low transferability of adversarial perturbations to other models, ensuring successful verification only when $M = M^{\text{copy}}$, even in cases where q_{trans} and q_{cloud} are unknown.

V. Conclusion and Future Work

This paper proposed a robust ownership verification framework for DNN models that remains effective under JPEG compression. Extending our previous work on probability-controlled adversarial attacks, we introduced I-FDGSM-AJ, which integrates JPEG simulation into the optimization process. To address unknown compression settings, we further proposed an adaptive method to estimate suitable generation parameters. Experimental results showed that the proposed approach enables reliable verification under both single and double JPEG compression, outperforming previous methods in lossy environments. This highlights its applicability to practical MLaaS scenarios where image degradation is common.

Future work includes extending the framework to other transformations such as resizing, format conversion (e.g., WebP), and defense-aware settings. While this study assumes aligned JPEG compression, the proposed method is designed to be adaptable and may be extended to more complex scenarios with additional development.

Acknowledgment

This study was supported by the JSPS KAKENHI(25K15225), JST SICORP (JPMJSC20C3), JST CREST (JPMJCR20D3), Japan.

References

[1] Y. Li, H. Wang, and M. Barni, "A survey of Deep Neural Network watermarking techniques," *Neurocomputing*, vol. 461, pp. 171–193, 2021.

[2] Y. Sun, T. Liu, P. Hu, *et al.*, "Deep intellectual property protection: A survey," *arXiv preprint arXiv:2304.14613*, 2023.

[3] J. Guo, A. Li, and C. Liu, "AEVA: Black-box backdoor detection using adversarial extreme value analysis," in *International Conference on Learning Representations (ICLR)*, 2022.

[4] Y. Yu, S. Hu, Y. Xiao, *et al.*, "Scale-up: Scalable black-box input-level backdoor detection via analyzing predictions across scales," *arXiv preprint arXiv:2302.03251*, 2023.

[5] T. Sano, M. Kuribayashi, M. Sakai, S. Isobe, and E. Koizumi, "Ownership verification of dnn models using white-box adversarial attacks with specified probability manipulation," *arXiv preprint arXiv:2505.17579*, 2025.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2015.

[7] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of JPG compression on adversarial images," *arXiv preprint arXiv:1608.00853*, 2016.

[8] X. Wang, X. Li, W. Zhang, C. Guo, J. Yan, and Z. Zhang, "Customized watermarking for deep neural networks via label distribution perturbation," *arXiv preprint arXiv:2208.05477*, 2022.

[9] Y. Adi, C. Baum, M. Cissé, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *Proceedings of the 27th USENIX Security Symposium*, 2018, pp. 1615–1631.

[10] X. Cao, J. Jia, and N. Z. Gong, "IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, ACM, 2021, pp. 14–25.