MATURITY-GUIDED SUPERVISION SHAPING FOR DATA-FREE MODEL EXTRACTION DEFENSE

Anonymous authors
Paper under double-blind review

ABSTRACT

Publicly accessible black-box models are vulnerable to model extraction attacks: by repeatedly querying the prediction interface and collecting its outputs, an attacker can train a substitute model with similar functionality. Existing defenses typically weaken clone training through output modification, randomized responses, query monitoring, simulated attack queries, or dynamic execution. Although effective, their defense signals often come from outside the model itself, making them sensitive to changes in query strategy, repeated-query smoothing, or deterministic-service requirements, which can lead to unstable defense performance or inconsistent service behavior. To avoid such external dependence, we propose Maturity-Guided Supervision Shaping (MGSS), a defense that derives its signal from the victim model's own inference process. MGSS exploits how class evidence matures across semantic depth and uses the resulting cross-stage maturity discrepancy as a victim-internal shaping signal. Internally aligned evidence supports utility-preserving responses, whereas stronger discrepancy steers the exposed supervision toward a less extractable direction, making it harder for the attacker to recover the original decision rule. MGSS freezes the victim backbone, fine-tunes only a lightweight module to adjust the final output, and preserves a deterministic fixed inference path without query history, randomized responses, or simulated attack queries. Experiments show that MGSS reduces clone accuracy under both soft-label and hard-label model extraction across multiple datasets and clone architectures. In the CIFAR-10 soft-label DFME setting, MGSS reduces the average accuracy of three clone architectures from 78.83% to 37.89%, while maintaining 94.46% clean victim accuracy.

1 INTRODUCTION

Black-box prediction services are valuable assets in machine-learning-as-a-service systems (Lowd & Meek, 2005; Tramèr et al., 2016; Oliynyk et al., 2023). Their returned probabilities or labels are intended to answer user queries, but in model extraction they can also be reused as training supervision: an adversary repeatedly queries the target model, collects input–output pairs, and trains a substitute model to mimic the victim (Tramèr et al., 2016; Papernot et al., 2017; Orekondy et al., 2019; Jagielski et al., 2020; Pal et al., 2020). Recent data-free extraction studies show that, even without the victim's training data, attackers can train substitute models using either probability outputs or only top-1 labels (Truong et al., 2021; Kariyappa et al., 2021b; Wang, 2021; Sanyal et al., 2022). Thus, defending against model extraction is not merely about hiding parameters, but about preserving normal predictive utility while reducing the cloning value of the supervision exposed by the prediction interface.

Existing active defenses weaken this exposed supervision by perturbing outputs, introducing randomness, using watermarking, simulating attack or OOD queries, or changing the execution path (Orekondy et al., 2020; Kariyappa et al., 2021a; Mazeika et al., 2022; Tang et al., 2024; Wang et al., 2023; 2024b;a; Luan et al., 2025). While effective, these defenses typically derive signals from mechanisms outside the frozen victim's inference process. This raises a simpler question: if the attacker exploits the model's returned supervision, can the defender use the victim model's process of forming that supervision to decide whether the returned output should be preserved or reshaped?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

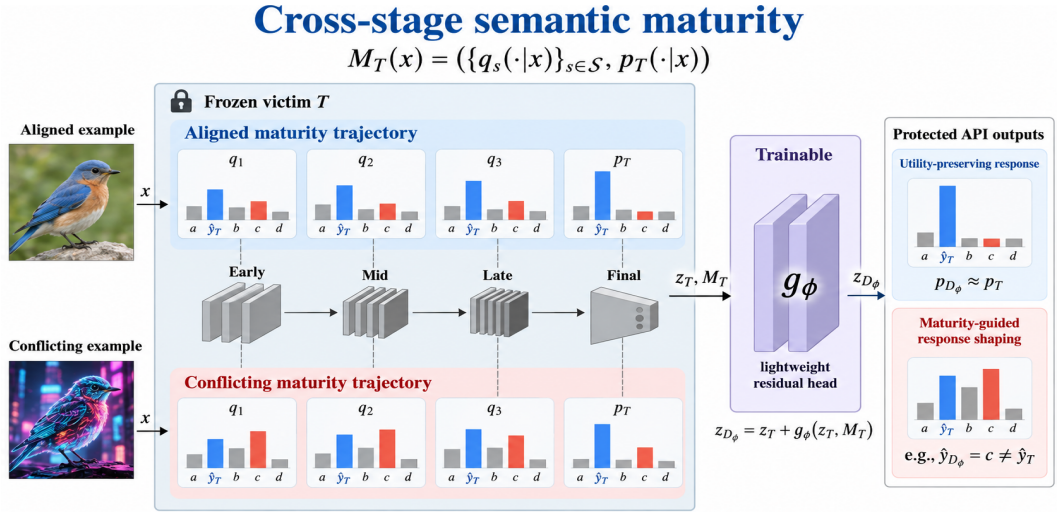

Figure 1: Overview of Maturity-Guided Supervision Shaping (MGSS). Prototype-based stage-wise class support q_1, q_2, q_3 and the final victim posterior p_T form the maturity trajectory $M_T(x)$. Aligned and conflicting examples illustrate representative patterns along a continuous maturity-discrepancy spectrum. A single shared residual head g_ϕ, the only trainable module, uses (z_T, M_T) to shape the exposed logits while retaining a deterministic fixed inference path.

Our key observation is that a deep model's prediction is formed progressively across internal stages. For the same input, early, middle, and late intermediate stages provide prototype-based class-support evidence, while the final classifier produces the victim posterior $p_T(\cdot \mid x)$. Together, these signals describe how semantic support matures toward the final prediction. This view is informed by work on early exits (Teerapittayanon et al., 2016; Kaya et al., 2019; Luan et al., 2025) and the information-bottleneck objective (Alemi et al., 2017). The degree to which intermediate support aligns with the final posterior varies across inputs. More internally aligned evidence supports utility-preserving responses, while stronger discrepancy can guide supervision shaping away from the original victim boundary.

We propose *Maturity-Guided Supervision Shaping* (MGSS), which avoids estimating or simulating attacker queries by deriving its shaping signal from the victim's own semantic maturation process. As illustrated in Figure 1, MGSS freezes the victim backbone and constructs a maturity trajectory from prototype-based class support at progressively deeper semantic stages together with the final victim posterior. The aligned and conflicting examples visualize representative patterns along this continuous maturity-discrepancy spectrum; they do not define a query detector or conditional execution path. MGSS instead uses the victim-internal trajectory to continuously shape the supervision exposed for each input. Service-preserving views anchor normal task behavior through the utility loss, whereas coverage views use maturity discrepancy to place more weight on final-suppressed intermediate support when constructing the defensive target.

This design requires only lightweight defensive tuning. Because the required signals are obtained from one forward pass of the frozen victim model, MGSS does not need to jointly train the backbone, an attacker-specific query generator, a random perturbation network, a watermark posterior network, or multiple branch classifiers. It trains only one lightweight residual head to map the internal maturity signal to a final logit adjustment, while preserving deterministic fixed-path inference.

We evaluate MGSS against soft-label and hard-label data-free extraction attacks across multiple datasets and substitute architectures, and further evaluate adaptive DFME variants on CIFAR-10. The results show that MGSS reduces clone accuracy while preserving the clean utility of the protected victim model.

Contributions.

- **We reformulate data-free model extraction defense as a utility-constrained supervision-shaping problem.** Instead of estimating the attacker query distribution, simulating specific attack strategies, or detecting query identities, MGSS directly reduces the learnable alignment between API-exposed supervision and the original victim decision rule. This formulation better matches realistic black-box API deployment constraints, where the defender does not know the attacker's query source, optimization objective, substitute architecture, or query strategy.
- **We identify cross-stage semantic maturity discrepancy as a victim-internal signal for model extraction defense.** This signal characterizes how class evidence develops across intermediate semantic stages and whether these stage-level evidences align or conflict with the final prediction. To the best of our knowledge, MGSS is the first defense to use the cross-stage semantic maturity dynamics of a frozen victim model to guide the construction of low-extractability supervision.
- **We propose Maturity-Guided Supervision Shaping (MGSS), a deterministic fixed-path defense based on the victim model's internal inference consistency.** MGSS freezes the victim backbone and trains only a lightweight residual head to shape the API output. It requires no query history, randomized response, attacker-specific query generator, model ensemble, early-exit routing, or test-time optimization, while preserving service utility and reducing model extractability.

2 Related Work

2.1 Model Extraction Attacks

Model extraction attacks train a substitute model from the responses of a black-box victim. Early and representative works study query-based model stealing, functionality extraction, fidelity-oriented extraction, and extraction beyond image classification (Lowd & Meek, 2005; Tramèr et al., 2016; Orekondy et al., 2019; Jagielski et al., 2020; Shen et al., 2022; Oliynyk et al., 2023). Representative data-based attacks start from an existing image pool or seed samples (Papernot et al., 2017; Orekondy et al., 2019; Pal et al., 2020; Jagielski et al., 2020). Data-free attacks do not require the victim's training data (Truong et al., 2021; Kariyappa et al., 2021b; Wang, 2021; Sanyal et al., 2022); DFME and MAZE generate queries without a surrogate dataset, while DFMS-HL studies the hard-label setting. Query source, API response type, and defense-aware adaptation are separate protocol choices; active query selection alone does not make an attack data-based. Soft-label attacks use probabilities, logits, or confidence scores, while hard-label attacks use only top-1 predictions. Because API probabilities act as soft supervision, model extraction is closely related to knowledge distillation and dark-knowledge transfer (Hinton et al., 2014; Nayak et al., 2019; Truong et al., 2021). Despite these differences, their common mechanism is the same: returned responses are converted into supervision for substitute-model training. MGSS targets this shared supervision-exposure channel rather than a query source, generator, or extraction loss.

2.2 Model Extraction Defenses

Existing defenses either detect, verify, delay, or actively disrupt extraction. Detection and monitoring methods identify suspicious users through query statistics, input similarity, OOD behavior, or information-leakage estimates (Juuti et al., 2019; Kesarwani et al., 2018; Pal et al., 2021), while delay and verification methods either increase query cost or provide post-hoc ownership evidence (Dziedzic et al., 2022; Adi et al., 2018; Uchida et al., 2017; Darvish Rouhani et al., 2019; Jia et al., 2021; Lv et al., 2024). These approaches provide auditing or deterrence capabilities. Active defenses modify the information exposed by the API through output perturbation, gradient redirection, ensembles, randomized inference, defensive training, active watermarking, dynamic early exits, or decision-region reshaping (Orekondy et al., 2020; Kariyappa & Qureshi, 2020; Kariyappa et al., 2021a; Mazeika et al., 2022; Tang et al., 2024; Wang et al., 2023; 2024b;a; Luan et al., 2025; Gong et al., 2024; Szyller et al., 2021; Xu et al., 2025; Wang et al., 2025). These methods show that weakening API-exposed supervision is an effective defense principle, but their shaping signals typically come from external mechanisms such as output-space perturbation objectives, randomized responses, simulated OOD or defense-shaping queries, ensembles, dynamic routing, or reconstructed

decision regions. MGSS differs in the source and use of the defense signal: it measures cross-stage class-support consistency inside the frozen victim model and uses this internal evidence to shape the final exposed logits through a single lightweight residual head. It therefore keeps a deterministic fixed inference path and does not require query history, randomized responses, an attacker-specific query generator, model ensembles, early-exit routing, or test-time optimization.

3 Preliminaries and Problem Formulation

This section formalizes the supervision channel exploited by query-based model extraction and the deployment constraints considered in this work. We then state the defense objective as utility-constrained supervision shaping, which motivates MGSS in Section 4.

3.1 Query-Based Extraction Threat

We consider image classification over input space $\mathcal{X}$ and C classes. Let T denote the pretrained victim model and let F denote the deployed prediction interface. Without defense, $F = T$; with defense, $F = D_\phi$, where D_ϕ is the protected API. For an input x, the deployed model computes logits $z_F(x)$, from which it obtains the posterior prediction $p_F(x) = \mathrm{softmax}(z_F(x))$ and the top-1 label $\hat{y}_F(x) = \arg\max_{c\in[C]} p_F(c \mid x)$. The API exposes either probabilities or top-1 labels:

$$a_F(x) \in \{p_F(x), \hat{y}_F(x)\}. \tag{1}$$

We use *exposed supervision* to emphasize that this returned response is the training target observed by the extractor. An attacker samples queries from an unknown distribution Q, collects API responses, and constructs

$$\mathcal{D}_Q(F) = \{(x_i, a_F(x_i))\}_{i=1}^N, \qquad x_i \sim Q. \tag{2}$$

The attacker then trains a substitute model S_θ by minimizing

$$\theta^\star(F, Q) = \arg\min_\theta \frac{1}{N} \sum_{i=1}^N \ell_{\mathrm{ext}}(S_\theta(x_i), a_F(x_i)), \tag{3}$$

where ℓ_{ext} can be a distillation loss for soft labels (Hinton et al., 2014; Truong et al., 2021) or a cross-entropy loss for hard labels (Sanyal et al., 2022). The query distribution Q is unknown to the defender and is not assumed to be out-of-distribution; it may be induced by synthetic generators, query augmentation, proxy data, actively selected samples, or their mixtures.

3.2 Defense Objective and Deployment Constraints

The defender has access to the pretrained victim model T and target-task data for preserving service utility, but does not know the attacker's query distribution, substitute architecture, optimization objective, query budget, or query strategy. The goal is to construct a protected API D_ϕ that preserves benign predictive utility—top-1 task behavior for hard-label APIs, with posterior fidelity and calibration additionally characterized for soft-label APIs—while reducing the quality of a substitute model trained from $\mathcal{D}_Q(D_\phi)$.

We view this goal as a utility-constrained supervision-shaping problem. The protected API should remain close to the original victim on service-preserving inputs, while exposing supervision that is less aligned with recovering the original victim boundary in regions where the victim's internal evidence is inconsistent. This formulation does not require detecting whether a query is malicious or estimating the attacker query distribution.

We focus on deterministic fixed-path deployment. The same input always receives the same response, and each query is processed independently. The defense does not rely on query history, user monitoring, randomized responses, attacker-specific query generators, model ensembles, early-exit routing, or test-time optimization. Under these constraints, the defense signal must come from the frozen victim model itself. MGSS therefore uses the consistency of stage-wise class support inside the victim model—which we call cross-stage semantic maturity discrepancy and define next—to shape the API-exposed supervision.

4 METHOD

This section first defines the victim-internal maturity trajectory used by MGSS, then presents the protected API parameterization, the maturity-guided defensive target, residual-head training, and the extractability-reduction mechanism.

Victim-internal maturity trajectory. Although the API exposes only the final response, the defender can inspect how class support evolves inside the frozen victim. For a small set of semantic stages $\mathcal{S}$, MGSS converts each stage feature into a class-support distribution (Snell et al., 2017)

$$q_s(c \mid x) = \frac{\exp(\mathrm{sim}(\bar{h}_s(x), \mu_{s,c})/\tau)}{\sum_{k=1}^{C} \exp(\mathrm{sim}(\bar{h}_s(x), \mu_{s,k})/\tau)}, \tag{4}$$

where $\bar{h}_s(x)$ is the pooled feature at stage s, $\mu_{s,c}$ is the class prototype for class c, and τ is a temperature. The victim-internal maturity trajectory is

$$M_T(x) = \big(\{q_s(\cdot \mid x)\}_{s\in\mathcal{S}}, p_T(\cdot \mid x)\big). \tag{5}$$

MGSS uses this trajectory to characterize how class support evolves across semantic depth and how intermediate support aligns with, or remains suppressed by, the final victim posterior.

4.1 MGSS OVERVIEW

Given a query x, the frozen victim produces final logits $z_T(x)$ and intermediate semantic representations from the selected stages. Fixed prototype-based readouts convert these representations into stage-wise class-support distributions, which together with $p_T(\cdot \mid x)$ form the maturity trajectory $M_T(x)$. MGSS attaches a lightweight residual head to the frozen victim and returns

$$z_{D_\phi}(x) = z_T(x) + g_\phi(z_T(x), M_T(x)), \qquad p_{D_\phi}(\cdot \mid x) = \mathrm{softmax}(z_{D_\phi}(x)). \tag{6}$$

All parameters of T remain frozen; only the residual-head parameters ϕ are trained. MGSS follows the design objective

$$\min_{\phi} \; R_{\mathrm{svc}}(D_\phi, T; P_{\mathrm{task}}) + \lambda_{\mathrm{def}} R_{\mathrm{shape}}(D_\phi, T; M_T, P_{\mathrm{cov}}), \tag{7}$$

where R_{svc} preserves service utility on P_{task}, and R_{shape} shapes supervision over the coverage distribution P_{cov}. We instantiate P_{task} with weakly augmented in-distribution samples and P_{cov} with non-attacker-specific strong augmentations from defender-side data and auxiliary public images when available. P_{cov} is used only offline to expose diverse maturity-discrepancy states; it neither contains attacker queries nor approximates an attacker query distribution. Thus, service views anchor normal behavior, while coverage views train the residual head to map victim-internal maturity states to shaping directions.

4.2 MATURITY DISCREPANCY AS A SHAPING SIGNAL

The trajectory $M_T(x)$ captures whether intermediate semantic supports agree with the final victim posterior. Before constructing the defensive target, we visualize whether this trajectory carries a measurable discrepancy signal in the frozen victim.

Figure 2 shows that generated extraction queries do not merely increase final uncertainty; they also enlarge the mismatch between stage-wise class support and the final posterior. MGSS therefore uses class-wise stage-final discrepancy rather than final confidence alone, and uses it continuously instead of thresholding it as a detector.

4.3 MATURITY-GUIDED DEFENSIVE TARGET

MGSS focuses on non-top-1 classes that are supported by intermediate stages but suppressed by the final prediction. We aggregate the intermediate supports and define final-suppressed support as

$$\begin{aligned} \bar{q}(c \mid x) &= \sum_{s\in\mathcal{S}} w_s q_s(c \mid x), \qquad \rho(x) = \sum_{c=1}^{C} r_c(x), \\ r_c(x) &= \big[\bar{q}(c \mid x) - p_T(c \mid x)\big]_+ \mathbf{1}[c \neq \hat{y}_T(x)]. \end{aligned} \tag{8}$$

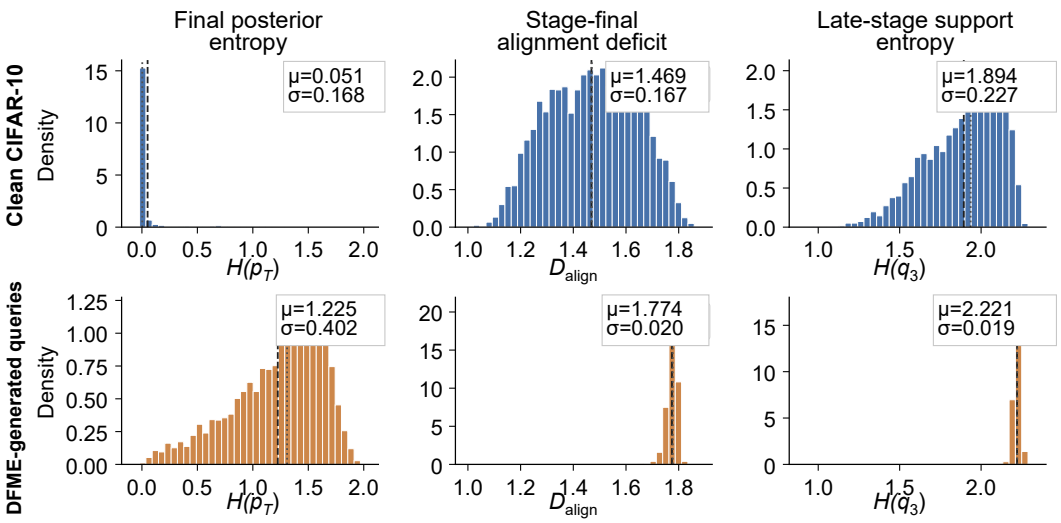

Figure 2: Offline maturity diagnostic on a frozen CIFAR-10 victim. DFME-generated diagnostic queries show higher final uncertainty, larger stage-final alignment deficit, and higher deep-stage uncertainty than clean inputs. This diagnostic only motivates the signal; MGSS does not use an online detector or assume this query distribution at deployment.

where $w_s \geq 0$ and $\sum_{s\in\mathcal{S}} w_s = 1$ are the stage aggregation weights, and $\hat{y}_T(x) = \arg\max_c p_T(c \mid x)$. The term $r_c(x)$ measures how strongly class c is supported by intermediate semantics but hidden from the final API response.

Let $\mathcal{C}^-(x) = \{c : c \neq \hat{y}_T(x)\}$. For a nonnegative vector v, $\mathrm{Norm}_{\mathcal{C}^-}(v)$ denotes normalization over the masked non-top-1 set. MGSS first defines the smoothed non-top-1 intermediate distribution and the final-suppressed support distribution:

$$q^-_{\mathrm{mid}}(\cdot \mid x) = \mathrm{Norm}_{\mathcal{C}^-}\big(\bar{q}(\cdot \mid x)\big), \qquad q_{\mathrm{sup}}(\cdot \mid x) = \begin{cases} \mathrm{Norm}_{\mathcal{C}^-}\big(r(x)\big), & \rho(x) > \epsilon, \\ q^-_{\mathrm{mid}}(\cdot \mid x), & \rho(x) \leq \epsilon. \end{cases} \tag{9}$$

The MGSS target is

$$q_{\mathrm{mgss}}(\cdot \mid x) = \alpha(x) q_{\mathrm{sup}}(\cdot \mid x) + \big(1 - \alpha(x)\big) q^-_{\mathrm{mid}}(\cdot \mid x), \qquad \alpha(x) = \frac{\rho(x)}{\rho(x) + \kappa}. \tag{10}$$

We use $\kappa > 0$ and $\epsilon \geq 0$. Since $\partial\alpha(x)/\partial\rho(x) = \kappa/(\rho(x) + \kappa)^2 > 0$, stronger maturity discrepancy places more weight on the final-suppressed support distribution, whereas weaker discrepancy places more weight on the smoothed non-top-1 intermediate distribution. This coverage target is still a counter-supervision target rather than a preservation target; normal task behavior is instead anchored by the service loss in Section 4.4. We additionally denote the dominant non-top-1 class of the MGSS target as

$$c^\star(x) = \arg\max_c q_{\mathrm{mgss}}(c \mid x). \tag{11}$$

This auxiliary definition is used only for characterization; residual-head training retains the full soft MGSS target in Eq. (13). At deployment, the hard-label API returns the top-1 prediction of the protected model,

$$\hat{y}_{D_\phi}(x) = \arg\max_c p_{D_\phi}(c \mid x), \tag{12}$$

rather than directly returning the offline counter-supervision target.

This target is not arbitrary label noise. It is derived from semantic evidence that the victim internally considered but did not expose as its final decision. A substitute trained on such supervision receives signals that are grounded in the victim's hidden trajectory but misaligned with the original victim boundary on internally inconsistent inputs.

4.4 Residual Head Training

Training uses two offline views with different roles. The service view preserves normal task utility, while the maturity-probing coverage view trains the residual head to expose the MGSS counter-supervision target:

$$\begin{aligned}\mathcal{L}_{\mathrm{MGSS}} = \mathbb{E}_{(x,y)\sim P_{\mathrm{task}}} \mathrm{CE}\big(p_{D_\phi}(A_w(x)), y\big) \\ + \lambda_{\mathrm{def}} \mathbb{E}_{u\sim P_{\mathrm{cov}}} \mathrm{CE}_{\mathrm{soft}}\big(q_{\mathrm{mgss}}(\cdot \mid A_s(u)), p_{D_\phi}(\cdot \mid A_s(u))\big),\end{aligned} \tag{13}$$

where

$$\mathrm{CE}_{\mathrm{soft}}(q, p) = -\sum_{c=1}^{C} q(c) \log p(c). \tag{14}$$

Here A_w and A_s denote service-preserving and maturity-probing augmentations, respectively, and λ_{def} controls the utility–defense trade-off. The service loss anchors task behavior, whereas coverage views use maturity discrepancy to vary the mixture weight between the two non-top-1 target distributions. Coverage inputs are used only to expose internal maturity states, without attack identities or attacker-query priors; T remains frozen and only ϕ is optimized.

4.5 Why Maturity-Guided Shaping Reduces Extractability

MGSS reduces extractability by making the API-exposed supervision less aligned with the victim decision rule on inputs with large maturity discrepancy. The attacker fits the protected API D_ϕ, not the victim T. Therefore, when MGSS induces protected–victim disagreement on the queried region, accurately fitting D_ϕ conflicts with recovering T. This disagreement is not random label noise; it is guided by final-suppressed intermediate support that is present in the victim's maturity trajectory.

Formally, for any query distribution Q, let $\hat{y}_S(x)$ denote the substitute model's top-1 prediction, and define the protected–victim disagreement rate $\delta_Q = \Pr_{x\sim Q}[\hat{y}_{D_\phi}(x) \neq \hat{y}_T(x)]$ and the substitute fitting error $\epsilon_Q = \Pr_{x\sim Q}[\hat{y}_S(x) \neq \hat{y}_{D_\phi}(x)]$. Then

$$\Pr_{x\sim Q}[\hat{y}_S(x) \neq \hat{y}_T(x)] \geq \delta_Q - \epsilon_Q. \tag{15}$$

Thus, a substitute that fits D_ϕ well can remain far from the original victim whenever MGSS creates sufficient protected–victim disagreement on the attacker's queried region. The bound is conditional on this queried-region disagreement and does not imply universal protection for every possible Q.

For soft-label extraction, the same mechanism applies: shaping $p_{D_\phi}(\cdot \mid x)$ away from $p_T(\cdot \mid x)$ makes distillation toward the protected API incompatible with matching the original victim posterior on inputs with large maturity discrepancy. Meanwhile, the service loss anchors task behavior on normal task inputs. Appendix C establishes the structural property of MGSS targets, the hard- and soft-label extraction conflicts, and residual-margin utility preservation.

5 Experiments

We evaluate the core extraction resistance, protected-victim utility, and adaptive extraction resistance in the main text, and provide additional utility, robustness, architecture-transfer, ablation, sensitivity, and efficiency analyses in the appendix.

5.1 Experimental Setup

Datasets and attacks. We evaluate MGSS on CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and MiniImageNet (Vinyals et al., 2016), using DFME (Truong et al., 2021) as the primary soft-label data-free attack and DFMS-HL (Sanyal et al., 2022) as a complementary hard-label data-free attack, following prior evaluations (Wang et al., 2023; 2024b;a). Additional CIFAR-10 MAZE (Kariyappa et al., 2021b) results are reported in Appendix E.1. The adaptive tests are DFME variants under the same data-free, soft-label setting and total query budget; we do not evaluate data-based extraction attacks. Soft-label APIs return probability vectors, whereas hard-label APIs return only top-1 labels.

Models and baselines. Unless otherwise specified, the victim is ResNet34-8x (He et al., 2016), and the substitutes are ResNet18-8x (He et al., 2016), MobileNetV2 (Sandler et al., 2018), and

DenseNet121 (Huang et al., 2017). We compare the undefended victim with MeCo (Wang et al., 2023), ACT (Wang et al., 2024b), AAUG (Wang et al., 2024a), and DNF (Luan et al., 2025); additional victim-architecture results are in Appendix E.4. Because AAUG has no official public implementation to our knowledge, we use its published values when protocols match and report N/A otherwise. Implementation details are in Appendix A.

Metrics and hardware. Soft-label query budgets are 20M on CIFAR-10 and 200M on CIFAR-100 and MiniImageNet; hard-label budgets are 8M, 10M, and 10M, respectively. For the adaptive DFME variants on CIFAR-10, repeated views and discarded candidates are counted within the same total query budget as vanilla DFME. Unless stated otherwise, results are five-run mean and standard deviation. Clone accuracy (lower is better) is the primary defense metric, and clean victim accuracy (higher is better) is the primary utility metric. Additional analyses are provided in Appendices B and D–G. Experiments used two RTX 3090 GPUs, 32GB RAM, and AMD EPYC 7542 CPUs, with each run using one GPU; training time, test-time memory, and parameter overhead are reported in Appendices H, J, and I.

5.2 Main Results: Defense against Data-Free Extraction

We evaluate MGSS as a standalone defense against data-free model extraction. Tables 1 and 2 report clone accuracy across the three datasets, where lower is better.

Soft-label extraction. Under DFME, MGSS reduces clone accuracy across all three datasets. On CIFAR-10, the ResNet18-8x clone drops from 87.36% without defense to 39.35%, with corresponding reductions for MobileNetV2 and DenseNet121, indicating reduced distillation utility across substitute architectures.

Hard-label extraction. Under DFMS-HL, MGSS also lowers clone accuracy across datasets and substitutes, showing that its protection extends beyond soft-label dark knowledge to label-only supervision.

Coverage. Across CIFAR-10, CIFAR-100, and MiniImageNet and all three substitute architectures, the results indicate that MGSS is not tied to a single dataset or substitute and support cross-stage maturity discrepancy as a useful shaping signal.

Table 1: Clone model accuracy on CIFAR-10 and CIFAR-100 with ResNet34-8x as the victim model. Lower is better.

		CIFAR-10			CIFAR-100		
Attack	Defense	ResNet18-8x	MobileNetV2	DenseNet121	ResNet18-8x	MobileNetV2	DenseNet121
DFMS-HL	Undefended	$84.67 \pm 1.90\%$	$79.28 \pm 1.87\%$	$68.87 \pm 2.08\%$	$72.57 \pm 1.28\%$	$62.71 \pm 1.68\%$	$63.58 \pm 1.79\%$
	MeCo	$76.86 \pm 2.09\%$	$71.22 \pm 1.87\%$	$62.33 \pm 2.01\%$	$59.30 \pm 1.70\%$	$55.32 \pm 1.65\%$	$56.80 \pm 1.86\%$
	ACT	$73.93 \pm 2.67\%$	$71.97 \pm 2.08\%$	$61.08 \pm 2.39\%$	$55.38 \pm 1.97\%$	$51.46 \pm 1.89\%$	$52.29 \pm 2.03\%$
	AAUG	$74.38 \pm 1.72\%$	$\mathbf{68.36 \pm 1.65\%}$	$62.36 \pm 1.81\%$	$45.68 \pm 1.42\%$	$46.98 \pm 1.83\%$	$49.56 \pm 1.91\%$
	DNF	$76.51 \pm 2.12\%$	$75.01 \pm 1.25\%$	$61.02 \pm 1.21\%$	$52.98 \pm 2.24\%$	$48.41 \pm 1.78\%$	$49.72 \pm 1.24\%$
	MGSS	$\mathbf{72.60 \pm 1.74\%}$	$70.47 \pm 2.08\%$	$\mathbf{57.53 \pm 2.09\%}$	$\mathbf{44.00 \pm 1.85\%}$	$\mathbf{46.09 \pm 1.92\%}$	$\mathbf{45.90 \pm 1.62\%}$
DFME	Undefended	$87.36 \pm 0.78\%$	$75.23 \pm 1.53\%$	$73.89 \pm 1.29\%$	$58.72 \pm 2.82\%$	$28.36 \pm 1.97\%$	$27.28 \pm 2.08\%$
	MeCo	$51.68 \pm 1.96\%$	$46.53 \pm 2.09\%$	$61.38 \pm 2.41\%$	$29.57 \pm 1.97\%$	$12.18 \pm 1.05\%$	$10.79 \pm 1.36\%$
	ACT	$46.57 \pm 2.83\%$	$40.32 \pm 2.96\%$	$49.25 \pm 2.67\%$	$23.95 \pm 2.38\%$	$10.09 \pm 2.53\%$	$6.26 \pm 2.38\%$
	AAUG	$46.82 \pm 1.79\%$	$45.17 \pm 1.93\%$	$40.91 \pm 1.89\%$	$21.31 \pm 2.03\%$	$10.38 \pm 2.16\%$	$9.87 \pm 2.19\%$
	DNF	$53.91 \pm 2.30\%$	$46.32 \pm 1.45\%$	$47.21 \pm 2.15\%$	$18.03 \pm 3.03\%$	$10.82 \pm 1.34\%$	$6.75 \pm 1.23\%$
	MGSS	$\mathbf{39.35 \pm 2.47\%}$	$\mathbf{37.54 \pm 2.96\%}$	$\mathbf{36.77 \pm 2.81\%}$	$\mathbf{17.39 \pm 2.58\%}$	$\mathbf{9.15 \pm 2.47\%}$	$\mathbf{6.01 \pm 2.61\%}$

Table 2: Clone model accuracy on MiniImageNet with ResNet34-8x as the victim model. Lower is better. AAUG is omitted because its code is not publicly available.

	DFMS-HL			DFME		
Defense	ResNet18-8x	MobileNetV2	DenseNet121	ResNet18-8x	MobileNetV2	DenseNet121
Undefended	$46.72 \pm 4.86\%$	$40.35 \pm 4.97\%$	$38.71 \pm 3.85\%$	$35.89 \pm 3.97\%$	$28.71 \pm 3.25\%$	$25.05 \pm 3.68\%$
MeCo	$39.23 \pm 4.83\%$	$35.81 \pm 4.69\%$	$32.30 \pm 4.56\%$	$23.29 \pm 3.83\%$	$17.83 \pm 3.67\%$	$16.73 \pm 3.88\%$
ACT	$33.28 \pm 4.57\%$	$31.61 \pm 4.38\%$	$27.32 \pm 4.75\%$	$19.81 \pm 3.57\%$	$15.36 \pm 3.56\%$	$12.95 \pm 3.28\%$
DNF	$38.50 \pm 5.49\%$	$37.63 \pm 4.77\%$	$30.25 \pm 4.92\%$	$18.26 \pm 3.81\%$	$15.90 \pm 3.59\%$	$15.63 \pm 4.19\%$
MGSS	$\mathbf{31.74 \pm 5.10\%}$	$\mathbf{30.56 \pm 4.93\%}$	$\mathbf{25.48 \pm 4.45\%}$	$\mathbf{15.63 \pm 3.89\%}$	$\mathbf{12.36 \pm 3.93\%}$	$\mathbf{10.51 \pm 3.82\%}$

5.3 Utility Preservation

Table 3: Victim model utility measured by clean test accuracy. Higher is better.

Method	CIFAR-10	CIFAR-100
Undefended	$94.91 \pm 0.37\%$	$76.71 \pm 1.25\%$
MeCo	$94.17 \pm 0.56\%$	$75.36 \pm 0.68\%$
ACT	$94.31 \pm 0.75\%$	$\mathbf{75.78 \pm 0.73}\%$
AAUG	$94.21 \pm 0.56\%$	$75.73 \pm 0.93\%$
DNF	$94.34 \pm 0.07\%$	$75.54 \pm 0.67\%$
MGSS	$\mathbf{94.46 \pm 0.31}\%$	$75.33 \pm 0.95\%$

Table 3 reports benign top-1 utility; MGSS reaches $94.46 \pm 0.31\%$ on CIFAR-10 and $75.33 \pm 0.95\%$ on CIFAR-100. For soft-label APIs, Appendix B additionally reports posterior fidelity and calibration.

5.4 Adaptive DFME and Additional Evaluations

Adaptive DFME variants. On CIFAR-10, we evaluate DFME-Smooth (multi-view response averaging), CA-DFME (multi-view consistency), and DFME-ISF (stability/disagreement-based candidate selection). All variants use the vanilla DFME total query budget, counting repeated views and discarded candidates (Appendix E.3).

Table 4: Clone model accuracy under vanilla DFME and adaptive DFME variants on CIFAR-10 with ResNet34-8x as the victim model. Results are five-run mean $\pm$ standard deviation; lower is better.

Defense	ResNet18-8x	MobileNetV2	DenseNet121
Undefended + DFME	$87.36 \pm 0.78\%$	$75.23 \pm 1.53\%$	$73.89 \pm 1.29\%$
MGSS + DFME	$39.35 \pm 2.47\%$	$37.54 \pm 2.96\%$	$36.77 \pm 2.81\%$
MGSS + DFME-Smooth	$48.19 \pm 2.91\%$	$45.19 \pm 3.07\%$	$46.83 \pm 2.82\%$
MGSS + CA-DFME	$46.94 \pm 3.56\%$	$42.85 \pm 2.91\%$	$46.77 \pm 2.67\%$
MGSS + DFME-ISF	$50.73 \pm 2.51\%$	$51.68 \pm 3.19\%$	$45.42 \pm 2.38\%$

Table 4 shows partial recovery: mean clone accuracy increases from 37.89% under vanilla DFME to 46.74%, 45.52%, and 49.28% under the three adaptive variants, respectively, but remains below the 78.83% undefended mean.

Maturity-discrepancy diagnostic. Appendix D shows higher stage-final agreement for clean inputs than for strong augmentations, proxy data, or DFME queries, while random-network controls rule out a purely architectural artifact.

5.5 Ablation and Sensitivity Analyses

Appendix F ablates the core MGSS components, and Appendix G analyzes defense-strength and coverage-source sensitivity under a fixed validation-utility constraint.

6 Conclusion

We presented *Maturity-Guided Supervision Shaping* (MGSS), which uses cross-stage evidence consistency in a frozen victim to deterministically shape API supervision without query detection, randomized responses, attack-query simulation, or dynamic execution. A lightweight residual head weakens the alignment between exposed responses and the original victim boundary on maturity-inconsistent inputs while preserving fixed-path inference. Experiments across soft- and hard-label extraction, datasets, and substitute architectures show reduced clone accuracy with maintained victim utility, supporting victim-internal evidence maturity as a useful signal for reducing black-box model extractability.

AI USE STATEMENT

Generative AI tools were used during the writing process to assist with language polishing, grammar checking, wording refinement, and minor formatting adjustments. All technical content, research claims, experimental results, analyses, and references were independently reviewed and verified by the authors. The authors take full responsibility for the accuracy and completeness of the manuscript.

REPRODUCIBILITY STATEMENT

The experimental setup, including the datasets, victim and substitute model architectures, attack settings, query budgets, evaluation metrics, and hardware configuration, is specified in Section 5.1. The appendices provide additional details on MGSS implementation and baseline assets, theoretical derivations, posterior-level utility, diagnostic analyses, adaptive-attack details, architecture-transfer evaluations, ablation studies, sensitivity analyses, and computational and resource overhead. These materials collectively document the experimental protocols and implementation choices needed to facilitate independent reproduction and verification of the reported results.

ETHICS STATEMENT

This work studies defenses against model extraction attacks in controlled machine-learning benchmark settings. The experiments use established benchmark datasets and previously published model extraction methods for defensive evaluation, and do not involve human subjects or private or sensitive personal data. Model extraction techniques are dual-use and may be misused to obtain unauthorized copies of proprietary machine-learning models. The purpose of this work is defensive: to better understand and reduce such risks by developing and evaluating mechanisms that limit the extractability of deployed models while preserving legitimate service utility. The work does not introduce a new model extraction attack, and the attack methods used in the evaluation are drawn from prior published research. Third-party datasets, software, and baseline implementations are used according to their stated licenses or usage terms where applicable, with relevant asset and license information documented in the appendix. The authors acknowledge the potential security implications of model extraction research and intend the presented techniques to support the protection of deployed machine-learning services.

REFERENCES

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *USENIX Security Symposium*, 2018.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. DeepSigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In *International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Adam Dziedzic, Muhammad Ahmad Kaleem, Yu Shen Lu, and Nicolas Papernot. Increasing the cost of model extraction with calibrated proof of work. In *International Conference on Learning Representations*, 2022.

Xueluan Gong, Rubin Wei, Ziyao Wang, Yuchen Sun, Jiawen Peng, Yanjiao Chen, and Qian Wang. Beowulf: Mitigating model extraction attacks via reshaping decision regions. In *ACM SIGSAC Conference on Computer and Communications Security*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning Workshop*, 2014. URL `https://arxiv.org/abs/1503.02531`.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

Matthew Jagielski, Nicholas Carlini, David Berthelot, Alexey Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *USENIX Security Symposium*, 2020.

Hengrui Jia, Christopher A. Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. Entangled watermarks as a defense against model extraction. In *USENIX Security Symposium*, 2021.

Mika Juuti, Sebastian Szyller, Samuel Marchal, and N. Asokan. PRADA: Protecting against DNN model stealing attacks. In *IEEE European Symposium on Security and Privacy*, 2019.

Sanjay Kariyappa and Moinuddin K. Qureshi. Defending against model stealing attacks with adaptive misinformation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Sanjay Kariyappa, Atul Prakash, and Moinuddin K. Qureshi. Protecting DNNs from theft using an ensemble of diverse models. In *International Conference on Learning Representations*, 2021a.

Sanjay Kariyappa, Atul Prakash, and Moinuddin K. Qureshi. MAZE: Data-free model stealing attack using zeroth-order gradient estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021b.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *International Conference on Machine Learning*, 2019.

Manish Kesarwani, Bhaskar Mukhoty, Vijay Arya, and Sameep Mehta. Model extraction warning in MLaaS paradigm. In *Annual Computer Security Applications Conference*, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005.

Siyu Luan, Zhenyi Wang, Li Shen, Zonghua Gu, Chao Wu, and Dacheng Tao. Dynamic neural fortresses: An adaptive shield for model extraction defense. In *International Conference on Learning Representations*, 2025.

Peizhuo Lv, Hualong Ma, Kai Chen, Jiachen Zhou, Shengzhi Zhang, Ruigang Liang, Shenchen Zhu, Pan Li, and Yingjun Zhang. MEA-Defender: A robust watermark against model extraction attack. In *IEEE Symposium on Security and Privacy*, 2024.

Mantas Mazeika, Bo Li, and David Forsyth. How to steer your adversary: Targeted and efficient model stealing defenses with gradient redirection. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 2022.

Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, 2019.

Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. I know what you trained last summer: A survey on stealing machine learning models and defences. *ACM Computing Surveys*, 55(14s):1–41, 2023.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Prediction poisoning: Towards defenses against DNN model stealing attacks. In *International Conference on Learning Representations*, 2020.

Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. ActiveThief: Model extraction using active learning and unannotated public data. In *AAAI Conference on Artificial Intelligence*, 2020.

Soham Pal, Yash Gupta, Aditya Kanade, and Shirish Shevade. Stateful detection of model extraction attacks. *arXiv preprint arXiv:2107.05166*, 2021.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM Asia Conference on Computer and Communications Security*, 2017.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Sunandini Sanyal, Sravanti Addepalli, and R. Venkatesh Babu. Towards data-free model stealing in a hard label setting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Yun Shen, Xinlei He, Yufei Han, and Yang Zhang. Model stealing attacks against inductive graph neural networks. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1175–1192. IEEE, 2022. doi: 10.1109/SP46214.2022.9833607.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Sebastian Szyller, Buse Gul Atli, Samuel Marchal, and N. Asokan. DAWN: Dynamic adversarial watermarking of neural networks. In *ACM Multimedia*, 2021.

Minxue Tang, Anna Dai, Louis DiValentin, Aolin Ding, Amin Hass, Neil Zhenqiang Gong, Yiran Chen, and Hai Li. ModelGuard: Information-theoretic defense against model extraction attacks. In *USENIX Security Symposium*, 2024.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *International Conference on Pattern Recognition*, 2016.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In *USENIX Security Symposium*, 2016.

Jean-Baptiste Truong, Pratyush Maini, Robert J. Walls, and Nicolas Papernot. Data-free model extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *ACM International Conference on Multimedia Retrieval*, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

Yixu Wang, Tianle Gu, Yan Teng, Yingchun Wang, and Xingjun Ma. HoneypotNet: Backdoor attacks against model extraction. In *AAAI Conference on Artificial Intelligence*, 2025.

Zhenyi Wang, Li Shen, Tongliang Liu, Tiehang Duan, Yanjun Zhu, Donglin Zhan, David Doermann, and Mingchen Gao. Defending against data-free model extraction by distributionally robust defensive training. In *Advances in Neural Information Processing Systems*, 2023.

Zhenyi Wang, Li Shen, Junfeng Guo, Tiehang Duan, Siyu Luan, Tongliang Liu, and Mingchen Gao. Training a secure model against data-free model extraction. In *European Conference on Computer Vision*, 2024a.

Zhenyi Wang, Yihan Wu, and Heng Huang. Defense against model extraction attack by Bayesian active watermarking. In *International Conference on Machine Learning*, 2024b.

Zi Wang. Zero-shot knowledge distillation from a decision-based black-box model. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10675–10685, 2021.

Yixiao Xu, Binxing Fang, Rui Wang, Yinghai Zhou, Shouling Ji, Yuan Liu, Mohan Li, and Zhihong Tian. Neural Honeytrace: A robust plug-and-play watermarking framework against model extraction attacks. *arXiv preprint arXiv:2501.09328v1*, 2025. URL `https://arxiv.org/abs/2501.09328v1`.

A Implementation and Baseline Details

A.1 MGSS Implementation Details

All victim-model parameters are kept frozen, and only the lightweight residual head is optimized. For the default ResNet34-8x victim, MGSS extracts intermediate representations from layer2, layer3, and layer4 as three semantically ordered stages. Class prototypes are constructed by aggregating normalized stage features from clean training samples, and stage-level class support is computed using cosine similarity with temperature $\tau = 0.07$. The stage features, original logits, and maturity descriptors are projected to 128-, 64-, and 32-dimensional representations, respectively, and a two-layer MLP with hidden dimension 256 produces the final logit residual. The output layer is zero-initialized so that the protected model starts from the original victim prediction. Training uses AdamW with an initial learning rate of 10^{-3}, weight decay 10^{-4}, cosine learning-rate scheduling, gradient clipping at 5, batch size 256, and 10 training epochs. The default defense weight is $\lambda_{\text{def}} = 0.05$. Service views use routine random cropping and horizontal flipping, while coverage views use stronger transformations to expose a broader range of maturity-discrepancy states.

A.2 Baselines and Assets

For baseline comparisons, we match the dataset, attack protocol, query budget, victim/substitute architectures, and evaluation metrics whenever possible. When an official or author-released implementation is available, we use it as the basis for reproduction. When no official implementation is available, published values are used only when the experimental protocol is directly comparable; otherwise the corresponding result is marked as unavailable.

Existing assets and licenses. For reproducibility, Table 5 summarizes the public datasets, standard libraries, and attack/defense implementations used or referenced in our experiments. When an official or author-released implementation is available, we use it as the basis for reproduction. For repositories without an explicit license file or license field, we report the license status as not explicitly specified. We do not redistribute third-party datasets, pretrained checkpoints, or external baseline repositories.

Table 5: Existing assets used in this work. "Not explicitly specified" means that no explicit license file or license field was found in the public repository page we accessed.

Asset	Type	Source / version	License / terms
CIFAR-10/100	Dataset	Original CIFAR website; public dataset source or standard dataset loaders. `https://www.cs.toronto.edu/~kriz/cifar.html`	Original dataset terms; no explicit license name found on the official dataset page. Used for research only; not redistributed.
MiniImageNet	Dataset	MiniImageNet split introduced by Matching Networks and commonly used with Ravi–Larochelle splits; images are sampled from ImageNet.	ImageNet non-commercial research / educational terms. Used for research only; not redistributed.
PyTorch / torchvision	Library	Standard dataset loaders, image transforms, and model components. `https://github.com/pytorch/vision`	BSD-3-Clause.
DFME	Attack code	Official public repository: `https://github.com/cake-lab/datafree-model-extraction`	GPL-3.0. Used only for baseline comparison; not redistributed.
DFMS-HL	Attack code	Official public repository: `https://github.com/val-iisc/Hard-Label-Model-Stealing`	Not explicitly specified. Used only for baseline comparison; not redistributed.
MAZE	Attack code	Public repository: `https://github.com/sanjaykariyappa/MAZE`	Not explicitly specified. Used only for baseline comparison; not redistributed.

Table 5: Existing assets used in this work (continued).

Asset	Type	Source / version	License / terms
RandP / P-poison	Defense code	Prediction Poisoning public repository: `https://github.com/tribhuvanesh/prediction-poisoning`	MIT. Used only for baseline comparison; not redistributed.
GRAD	Defense code	Gradient Redirection public repository: `https://github.com/mmazeika/model-stealing-defenses`	MIT. Used only for baseline comparison; not redistributed.
MeCo	Defense code	Author public repository: `https://github.com/joey-wang123/DFME-DRO`	Not explicitly specified. Used only for baseline comparison; not redistributed.
ACT	Defense code	Author public repository: `https://github.com/joey-wang123/Bayes-Active-Watermark`	Not explicitly specified. Used only for baseline comparison; not redistributed.
AAUG	Defense method	No official public implementation found. We use the published paper values when the dataset and protocol match, and mark the result as N/A otherwise.	No third-party code asset used for AAUG reproduction unless otherwise stated.
DNF	Defense code	Author public repository: `https://github.com/SYCodeShare/Dynamic-Neural-Fortresses`	Apache-2.0. Used only for baseline comparison; not redistributed.

B Additional Utility Results

B.1 MiniImageNet Utility

Table 6: Victim model utility on MiniImageNet measured by clean test accuracy. Higher is better. AAUG is marked unavailable because its code is not publicly released, so we cannot reproduce its MiniImageNet utility under our setting.

Method	MiniImageNet
Undefended	$60.57 \pm 1.67\%$
MeCo	$58.66 \pm 1.91\%$
ACT	$58.59 \pm 1.94\%$
AAUG	N/A (code not public)
DNF	$\mathbf{58.94 \pm 1.72}\%$
MGSS	$58.74 \pm 1.67\%$

B.2 Posterior-Level Utility Preservation

In addition to clean classification accuracy, we evaluate the posterior-level service utility of MGSS on benign inputs. This analysis uses matched original-victim and MGSS checkpoints and computes calibration and posterior-fidelity metrics sample by sample over the full benign test set. The five-run mean and standard deviation reported in Table 3 remain the primary utility estimate; the matched-checkpoint analysis here is intended to characterize probability-level changes rather than replace the main utility statistics.

Table 7: Posterior-level utility on benign test inputs using matched original-victim and MGSS checkpoints. Lower ECE, NLL, Brier score, KL divergence, and TV distance are better; higher Top-1 agreement indicates stronger prediction consistency.

Dataset	ECE Original / MGSS	NLL Original / MGSS	Brier Original / MGSS	Top-1 Agreement	KL	TV
CIFAR-10	0.0316 / 0.0312	0.2018 / 0.2355	0.0787 / 0.0903	97.49%	0.0554	0.0266
CIFAR-100	0.0959 / 0.1074	1.0022 / 1.0719	0.3485 / 0.3551	90.51%	0.1587	0.0971

On CIFAR-10, MGSS retains 97.49% Top-1 agreement with the original victim, with KL and TV distances of 0.0554 and 0.0266, respectively. On CIFAR-100, the corresponding values are 90.51%, 0.1587, and 0.0971. Although some calibration and likelihood metrics show moderate degradation, the results indicate that the probability outputs on benign inputs remain substantially aligned with those of the original victim.

C Theoretical Justification of MGSS

Let T denote the original victim, D_ϕ the protected API, and S a substitute. Their posterior distributions are p_T, p_{D_ϕ}, and p_S, with top-1 predictions $\hat{y}_T$, $\hat{y}_{D_\phi}$, and $\hat{y}_S$. Throughout, Q is an arbitrary attacker query distribution; we do not assume $Q = P_{\text{cov}}$. The goal is not to prove universal resistance for every attacker or query distribution. Rather, we characterize the conditional conflict that arises once MGSS induces protected–victim separation on a queried region: fitting the protected API then conflicts with recovering the original victim.

C.1 Structural Property of MGSS Targets

We assume $C \geq 2$, $w_s \geq 0$, $\sum_s w_s = 1$, $\kappa > 0$, and $\epsilon \geq 0$, as in Eqs. (8)–(10).

Proposition 1 (Victim-top-1 exclusion)**.** *For every input x, the MGSS target satisfies*

$$q_{\text{mgss}}(\hat{y}_T(x) \mid x) = 0.$$

Proof. The indicator in Eq. (8) gives $r_{\hat{y}_T(x)}(x) = 0$. Both q^-_{mid} and q_{sup} in Eq. (9) are normalized only on $\mathcal{C}^-(x) = \{c : c \neq \hat{y}_T(x)\}$, so both assign zero mass to $\hat{y}_T(x)$. Because $\alpha = \rho/(\rho + \kappa) \in [0, 1)$, their mixture in Eq. (10) also assigns zero mass to that class.

This establishes the directional property of the MGSS target; it does not by itself imply that the trained protected API changes its top-1 prediction.

C.2 Extraction Conflict

For classifiers A and B, define their disagreement on Q as

$$d_Q(A, B) = \Pr_{x \sim Q}[\hat{y}_A(x) \neq \hat{y}_B(x)].$$

Theorem 1 (Hard-label extraction conflict)**.** *For any T, D_ϕ, S, and Q,*

$$d_Q(S, T) \geq d_Q(D_\phi, T) - d_Q(S, D_\phi).$$

Proof. For every x,

$$\mathbf{1}[\hat{y}_S \neq \hat{y}_T] \geq \mathbf{1}[\hat{y}_{D_\phi} \neq \hat{y}_T] - \mathbf{1}[\hat{y}_S \neq \hat{y}_{D_\phi}].$$

Taking expectation over $x \sim Q$ proves the claim.

Writing $\delta_Q = d_Q(D_\phi, T)$ and $\epsilon_Q = d_Q(S, D_\phi)$ recovers Eq. 15:

$$d_Q(S, T) \geq \delta_Q - \epsilon_Q.$$

Thus, if MGSS has already induced protected–victim disagreement on the queried region, decreasing the substitute fitting error to D_ϕ does not imply recovery of T; instead, the substitute inherits the protected API's disagreement with T. The theorem does not assert that δ_Q is large for every Q.

For soft-label extraction, let

$$\mathrm{TV}(p,q) = \frac{1}{2}\|p-q\|_1, \qquad \Delta_Q(A,B) = \mathbb{E}_{x\sim Q}\,\mathrm{TV}(p_A(\cdot \mid x), p_B(\cdot \mid x)).$$

Theorem 2 (Soft-label extraction conflict)**.** *For any T, D_ϕ, S, and Q,*

$$\Delta_Q(S,T) \geq \Delta_Q(D_\phi,T) - \Delta_Q(S,D_\phi).$$

Proof. The total-variation triangle inequality gives $\mathrm{TV}(p_{D_\phi}, p_T) \leq \mathrm{TV}(p_{D_\phi}, p_S) + \mathrm{TV}(p_S, p_T)$ pointwise. Rearranging and taking expectation over Q proves the result.

Corollary 1 (KL-based distillation)**.** *Suppose KL divergence uses natural logarithms and define*

$$\epsilon_Q^{\mathrm{KL}} = \mathbb{E}_{x\sim Q}\,\mathrm{KL}\big(p_{D_\phi}(\cdot \mid x) \,\|\, p_S(\cdot \mid x)\big)\,.$$

Then

$$\Delta_Q(S,T) \geq \Delta_Q(D_\phi,T) - \sqrt{\epsilon_Q^{\mathrm{KL}}/2}.$$

Proof. Pinsker's inequality and Jensen's inequality give $\Delta_Q(S,D_\phi) \leq \sqrt{\epsilon_Q^{\mathrm{KL}}/2}$; substitution into Theorem 2 yields the claim.

Once the protected posterior is separated from the victim posterior, accurately fitting or distilling the protected posterior is incompatible with simultaneously eliminating the substitute–victim posterior discrepancy. The TV result is attack-objective agnostic; Corollary 1 specializes it to KL-based distillation.

C.3 Utility Preservation under Residual Shaping

Write the protected logits as

$$z_{D_\phi}(x) = z_T(x) + \Delta_\phi(x), \qquad \Delta_\phi(x) = g_\phi(z_T(x), M_T(x)).$$

For $t = \hat{y}_T(x)$, define the victim prediction margin

$$m_T(x) = z_T(t \mid x) - \max_{c\neq t} z_T(c \mid x).$$

Proposition 2 (Prediction stability under residual shaping)**.** *If*

$$m_T(x) > 2\|\Delta_\phi(x)\|_\infty,$$

then $\hat{y}_{D_\phi}(x) = \hat{y}_T(x)$.

Proof. For every $c \neq t$,

$$\begin{aligned} z_{D_\phi}(t \mid x) - z_{D_\phi}(c \mid x) &= z_T(t \mid x) - z_T(c \mid x) + \Delta_\phi(t \mid x) - \Delta_\phi(c \mid x) \\ &\geq m_T(x) - 2\|\Delta_\phi(x)\|_\infty > 0. \end{aligned}$$

Hence t remains the protected API's top-1 prediction.

Corollary 2 (Utility lower bound)**.** *For any labeled evaluation distribution P,*

$$\begin{aligned} \Pr_{x\sim P}[\hat{y}_{D_\phi}(x) \neq \hat{y}_T(x)] &\leq \Pr_{x\sim P}[m_T(x) \leq 2\|\Delta_\phi(x)\|_\infty], \\ \mathrm{Acc}_P(D_\phi) &\geq \mathrm{Acc}_P(T) - \Pr_{x\sim P}[m_T(x) \leq 2\|\Delta_\phi(x)\|_\infty]. \end{aligned}$$

Proof. Proposition 2 implies the first inequality. The second follows from $\mathrm{Acc}_P(T) - \mathrm{Acc}_P(D_\phi) \leq \Pr_P[\hat{y}_{D_\phi} \neq \hat{y}_T]$.

This is a sufficient prediction-stability condition. In the MGSS objective, the service term anchors normal task behavior; the margin result provides a sufficient condition under which the residual adjustment preserves the victim prediction.

Overall implication. These results provide a conditional defense–utility characterization of MGSS. Once supervision shaping induces a separation between D_ϕ and T on a queried region, a substitute that better fits D_ϕ cannot simultaneously eliminate its discrepancy from T, under both hard-label and soft-label extraction. Meanwhile, residual shaping preserves the victim prediction whenever the original prediction margin dominates the induced logit residual. The analysis characterizes the supervision-shaping mechanism; the usefulness of cross-stage maturity for determining where and how to shape is evaluated empirically through the diagnostic and ablation studies.

D Additional Evidence for Cross-Stage Maturity Consistency

Diagnostic goal. This appendix provides diagnostic evidence for the basic observation behind MGSS: a trained victim model exhibits stable cross-stage semantic maturity on normal inputs, while stronger perturbations, proxy inputs, and DFME-generated queries weaken this consistency. This experiment is not a defense-effectiveness evaluation and is not an online attack-query detection experiment. All statistics are computed on the frozen clean ResNet34-8x CIFAR-10 teacher before training the MGSS defended head.

For each input x, we inspect three intermediate semantic stages of the victim model:

$$\mathcal{S} = \{\text{layer2}, \text{layer3}, \text{layer4}\}.$$

For each stage $s \in \mathcal{S}$, we compute a prototype-based class-support distribution $q_s(\cdot|x)$ using class prototypes estimated from the clean training set, following the prototype-based classification principle (Snell et al., 2017). We then compare the top-1 class supported by each intermediate stage with the top-1 class of the final prediction distribution $p_T(\cdot|x)$.

We use the following stage-final top-1 agreement as the main diagnostic metric:

$$A_{\text{stage}}(x) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathbf{1}\left[\arg\max_c q_s(c|x) = \arg\max_c p_T(c|x)\right].$$

A higher A_{stage} indicates that intermediate semantic stages and the final classifier support the same class more consistently, suggesting a more stable semantic maturity path. A lower value indicates stronger disagreement across stages.

To avoid drawing conclusions from a single generator snapshot, we evaluate the evolving DFME generator checkpoints from epoch 1 to epoch 54 and summarize them into three consecutive phases: early, middle, and late. Each checkpoint contributes 10,000 generated queries, and the DFME-GAN phase rows report checkpoint-level averages over the specified epochs.

Table 8: Diagnostic evidence for cross-stage maturity consistency on a frozen clean CIFAR-10 victim before MGSS training. *Final conf.* and *Entropy* measure final-layer uncertainty. A_{stage} measures the average top-1 agreement between intermediate stages and the final prediction. For DFME-GAN phase rows, we report checkpoint-level averages over the specified epochs.

Query source	N	Final conf.	Entropy	A_{stage} (%)	Acc. (%)
Clean ID test set	10000	0.983 ± 0.070	0.050 ± 0.166	80.77	95.25
Weak augmentation	10000	0.982 ± 0.074	0.056 ± 0.179	80.40	94.64
Medium augmentation	10000	0.924 ± 0.150	0.222 ± 0.376	67.39	76.43
Strong augmentation	10000	0.899 ± 0.169	0.300 ± 0.433	64.44	67.72
CIFAR-100 related classes	4800	0.869 ± 0.177	0.373 ± 0.422	61.35	–
CIFAR-100 unrelated classes	5100	0.837 ± 0.190	0.466 ± 0.458	57.94	–
DFME-GAN early, ep1–12	12×10000	0.702	0.830	56.27	–
DFME-GAN middle, ep13–21	9×10000	0.656	0.933	50.98	–
DFME-GAN late, ep22–54	33×10000	0.685	0.855	53.04	–
Random net / Clean ID	10000	0.660 ± 0.146	0.798 ± 0.229	13.30	–
Random net / DFME-GAN	10000	0.665 ± 0.149	0.787 ± 0.239	13.75	–

Column descriptions. *Query source* denotes the input source used for offline diagnostics. N is the number of evaluated samples. For DFME-GAN phase rows, N denotes the number of checkpoints multiplied by the number of generated queries per checkpoint. *Final conf.* is the mean maximum final posterior probability, i.e., $\max_c p_T(c|x)$. *Entropy* is the entropy of the final prediction distribution $p_T(\cdot|x)$. A_{stage} is the average top-1 agreement between layer2/layer3/layer4 and the final prediction. *Acc.* denotes teacher accuracy when ground-truth labels are available. For random-network controls, final confidence and entropy are reported for completeness; the interpretation focuses on A_{stage}.

Observations. Table 8 shows that clean-ID samples have the most stable cross-stage semantic maturity path in the trained victim model. Clean ID obtains the highest stage-final agreement, with $A_{\text{stage}} = 80.77\%$, while weak augmentation remains close at 80.40%. This indicates that normal service-like inputs usually preserve consistent top-1 semantic support across intermediate stages and the final classifier.

As input perturbation becomes stronger, this consistency decreases substantially. Medium augmentation reduces A_{stage} to 67.39%, and strong augmentation further reduces it to 64.44%. Natural proxy queries show the same tendency: CIFAR-100 related and unrelated classes obtain 61.35% and 57.94%, respectively. Thus, stage-final maturity consistency is sensitive to both artificial perturbations and semantic distribution shift.

The evolving DFME generator produces even lower stage-final agreement. In the early phase, DFME-GAN queries already reduce A_{stage} to 56.27%, below standard strong augmentation. The middle phase produces the strongest disruption, reducing A_{stage} to 50.98% while also yielding high final-layer uncertainty. In the late phase, the agreement slightly recovers to 53.04%, but remains far below clean ID and standard augmentations. This suggests that DFME query synthesis tends to explore regions where the victim's intermediate semantic evidence and final prediction are less aligned.

The random-network controls rule out a structural artifact. A randomly initialized ResNet34-8x obtains only 13.30% stage-final agreement on clean-ID samples and 13.75% on DFME-GAN queries, which is close to random agreement in a ten-class task. In contrast, the trained clean victim obtains 80.77% agreement on clean-ID samples. Therefore, the high stage-final agreement observed on clean data is a learned semantic property of the trained victim model, rather than a consequence of the ResNet architecture or the prototype-based diagnostic procedure itself.

Takeaway. These diagnostics support the premise that cross-stage semantic maturity consistency exists in the frozen clean victim before MGSS training. Normal inputs follow a more stable maturity path, while stronger perturbations, semantic shifts, and DFME-generated queries disrupt this path to

different degrees. This provides diagnostic support for MGSS: the victim model contains internal, API-hidden maturity signals that can be used for supervision shaping without relying on an online attack-query detector.

E Additional Extraction Results

E.1 MAZE Results

Additional score-based extraction evaluation. Table 9 reports CIFAR-10 results under MAZE (Kariyappa et al., 2021b), using the same victim, substitute architectures, and defenses as in the main experiments.

Table 9: Clone model accuracy under MAZE on CIFAR-10 with ResNet34-8x as the victim model. Lower is better.

Defense	ResNet18-8x	MobileNetV2	DenseNet121
Undefended	$45.17 \pm 0.73\%$	$23.28 \pm 1.67\%$	$20.03 \pm 1.79\%$
MeCo	$21.89 \pm 2.07\%$	$18.75 \pm 2.11\%$	$17.95 \pm 1.46\%$
ACT	$18.09 \pm 2.31\%$	$16.57 \pm 2.06\%$	$17.21 \pm 1.97\%$
AAUG	$21.67 \pm 2.02\%$	$19.20 \pm 1.96\%$	$17.32 \pm 1.83\%$
DNF	$22.46 \pm 2.83\%$	$21.10 \pm 2.28\%$	$20.94 \pm 2.62\%$
MGSS	$\mathbf{15.68 \pm 1.83\%}$	$\mathbf{14.20 \pm 2.01\%}$	$\mathbf{14.90 \pm 1.95\%}$

MGSS achieves the lowest clone accuracy across all three substitute architectures, providing an additional evaluation consistent with the main DFME results.

E.2 Budget-Sensitivity Results

Evaluation under increasing query budgets. Clone accuracy under increasing DFME query budgets on CIFAR-10 with a ResNet34-8x victim and ResNet18-8x clone. MGSS maintains lower clone accuracy across budgets, suggesting that the protection is not limited to a small-query regime.

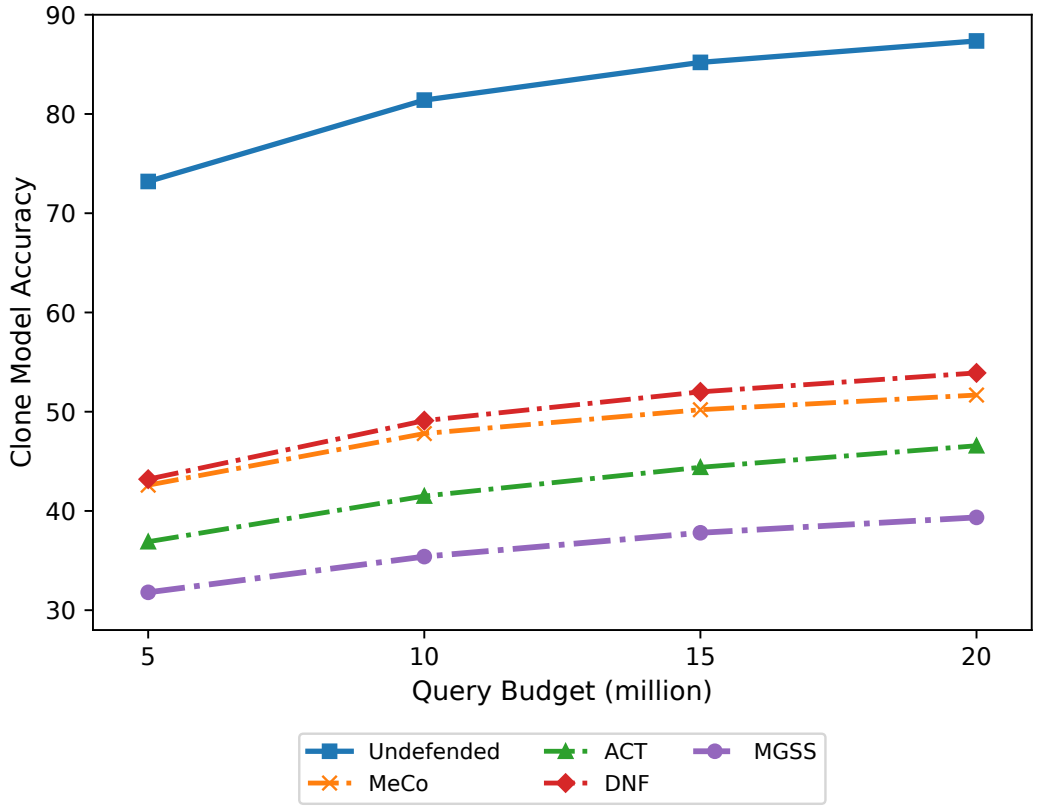

Figure 3: Clone model accuracy under increasing query budgets. The x-axis denotes the query budget in millions, and the y-axis denotes the clone model accuracy. As the query budget increases, the clone model generally obtains more supervision and its accuracy improves. Nevertheless, MGSS remains in the low-accuracy region across budgets and converges around the strongest defense group, indicating that Maturity-Guided Supervision Shaping continues to suppress extractable supervision even when more queries are available.

Discussion. Figure 3 shows that increasing the query budget improves the attacker's clone model, which is expected because additional API responses provide more supervision for substitute-model training. However, MGSS consistently maintains a substantially lower clone accuracy than the undefended API and remains competitive with recent randomized, defensive-training, and dynamic-inference baselines such as MeCo, ACT, and DNF. AAUG is not included in this budget-sensitivity comparison because its code is not publicly available, so we cannot reproduce the corresponding curve. MGSS also preserves its deterministic deployment contract: the victim backbone is frozen, only a lightweight defended head is trained, and inference does not rely on query history, randomized sampling, test-time optimization, or model ensembles. This result supports the central claim that MGSS reduces the learnability of the exposed supervision rather than merely perturbing the final posterior vector.

E.3 Adaptive DFME Variants

The complete results for the adaptive DFME variants on CIFAR-10 with a ResNet34-8x victim are reported in Table 4. Here we provide the corresponding attack descriptions and evaluation protocol.

CA-DFME augments the zeroth-order generator objective with multi-view consistency to encourage informative queries with stable API responses. DFME-ISF generates a larger synthetic candidate pool and selects queries using both multi-view stability and victim–clone disagreement. These attacks use only synthetic queries, API-returned probabilities, and the attacker's clone; they do not access MGSS

intermediate features, maturity signals, defense-training data, or residual-head parameters. Repeated views and discarded candidates are counted within the same total query budget as vanilla DFME.

E.4 Victim-Architecture Generalization

To examine whether MGSS depends on the ResNet34-8x residual structure or its specific stage partition, we transfer the defense to ViT (Dosovitskiy et al., 2021), MobileNetV2, and DenseNet121 victims. For convolutional victims, we select three architecture-native readout locations ordered by semantic depth. For the 12-block ViT, blocks 4, 8, and 12 are used as the three semantic readouts. Apart from this architecture-specific readout selection, the prototype construction, stage-wise class-support computation, maturity-discrepancy formulation, and additive logit-residual pipeline remain unchanged. The victim backbone is frozen in all cases, only the lightweight residual head is trained, and the attacker uses a ResNet18-8x substitute under the same DFME protocol and query budget.

Table 10: Supplementary victim-architecture transfer results under DFME. These experiments are single-run evaluations with a ResNet18-8x substitute; lower clone accuracy is better.

Victim	Original Acc. (%)	MGSS Acc. (%)	Undefended Clone (%)	MGSS Clone (%)
ViT	90.46	90.20	55.99	27.33
MobileNetV2	92.63	92.28	80.67	39.05
DenseNet121	94.59	93.62	77.73	26.77

MGSS substantially reduces DFME clone accuracy for all three tested non-default victim architectures, while the protected-victim accuracy drops by 0.26, 0.35, and 0.97 percentage points for ViT, MobileNetV2, and DenseNet121, respectively. These results indicate that MGSS is not specific to the ResNet34-8x victim in the tested architectures. However, MGSS still requires semantically meaningful ordered intermediate representations, so these results should not be interpreted as universal architecture independence.

F Ablation Details

This section describes the ablation variants used in Table 11. All variants are evaluated under the same setting as the main data-free experiment: CIFAR-10, soft-label DFME, ResNet34-8x victim model, and ResNet18-8x clone model. The query budget, substitute training protocol, evaluation set, and random seeds are kept the same across all variants. The victim backbone remains frozen in all MGSS variants.

Full MGSS. Full MGSS uses cross-stage semantic maturity discrepancy to estimate the shaping strength and uses maturity-guided semantic evidence to determine the shaping direction. The protected API is deterministic and performs a single fixed-path forward pass at inference time.

w/o cross-stage maturity. This variant removes the cross-stage maturity discrepancy signal. Instead of using intermediate-to-final maturity disagreement, the shaping strength is estimated only from the final-layer output confidence:

$$\rho_{\text{final}}(x) = 1 - \max_{c} p_T(c \mid x),$$

where $p_T(c \mid x)$ denotes the posterior probability of the frozen victim model. This variant tests whether final-layer confidence alone is sufficient for supervision shaping.

w/o maturity-guided direction. This variant keeps the shaping strength estimated from cross-stage maturity discrepancy, but removes the maturity-guided semantic direction. Instead of using intermediate semantic evidence to select the residual shaping direction, the target direction is chosen as the second-largest class in the final-layer logits:

$$c_{\text{alt}}(x) = \arg \max_{c \neq \hat{y}_T(x)} z_T(c \mid x),$$

where $z_T(c \mid x)$ is the final logit of the frozen victim and $\hat{y}_T(x)$ is its top-1 prediction. This variant tests whether MGSS is more effective than a generic final-logit-based perturbation direction.

Hard-gated shaping. This variant replaces the continuous shaping strength in MGSS with a hard threshold. Given the maturity discrepancy score $\rho(x)$, the shaping strength is defined as

$$\alpha_{\text{gate}}(x) = \begin{cases} 0, & \rho(x) \le \tau_{\text{gate}}, \\ \alpha_{\max}, & \rho(x) > \tau_{\text{gate}}, \end{cases}$$

where τ_{gate} is selected on the held-out validation split and $\alpha_{\max}$ is the maximum shaping strength used by MGSS. This variant tests whether continuous maturity-conditioned modulation provides a better defense–utility trade-off than a binary detection-and-shaping rule.

Evaluation protocol. For each variant, we report clone model accuracy and protected victim accuracy. Clone accuracy measures the effectiveness of model extraction, while victim accuracy measures the benign-service utility of the protected API. A useful ablation variant should therefore be interpreted through both metrics rather than clone accuracy alone.

Table 11: Ablation study on CIFAR-10 under soft-label DFME. Lower clone accuracy and higher victim accuracy are better.

Method	Clone Acc. ↓	Victim Acc. ↑
Undefended	87.36 ± 0.78	94.91 ± 0.37
MGSS	39.35 ± 2.47	94.46 ± 0.31
w/o Cross-stage Maturity	80.59 ± 5.06	95.06 ± 0.15
w/o Maturity-guided Direction	47.81 ± 2.60	94.17 ± 0.29
Hard-gated Shaping	55.38 ± 3.92	93.89 ± 0.74

Table 11 evaluates whether the main components of MGSS are necessary. The comparison is designed to separate three effects: whether cross-stage maturity provides useful information beyond final confidence, whether the shaping direction should be guided by intermediate semantic evidence, and whether continuous modulation is preferable to hard gating. These ablations directly test whether MGSS improves the defense–utility trade-off through maturity-guided response shaping rather than through generic output distortion.

G Sensitivity Analyses

G.1 Utility–Defense Trade-off under Different Shaping Strengths

We evaluate the effect of the defense weight λ_{def} under the same CIFAR-10 soft-label DFME setting. The purpose of this study is not to show that MGSS is insensitive to this hyperparameter. Instead, λ_{def} serves as an explicit utility–defense trade-off knob. A larger λ_{def} applies stronger maturity-guided shaping, which can further reduce the clone model accuracy but may also affect benign service utility. A smaller λ_{def} better preserves the protected victim accuracy, but it leaves more informative supervision available to the clone model.

For the main experiments, we use a balanced operating point selected under a fixed utility constraint on validation data. Specifically, we choose the largest λ_{def} that keeps the protected victim accuracy within the allowed validation utility drop, and then report the corresponding extraction resistance. This protocol reflects a practical deployment setting: the service provider first enforces a service-utility requirement and then maximizes extraction resistance within the allowed utility budget.

Table 12: Utility–defense trade-off under different defense weights λ_{def} on CIFAR-10 under the soft-label DFME attack. Lower clone accuracy indicates stronger extraction resistance, while higher victim accuracy indicates better benign service utility.

λ_{def}	Clone Acc. $\downarrow$	Victim Acc. $\uparrow$
0.01	45.92 ± 2.83	94.81 ± 0.17
0.05	39.35 ± 2.47	94.46 ± 0.31
0.10	35.09 ± 2.42	93.89 ± 0.21
0.50	33.81 ± 4.13	92.81 ± 0.34

Table 12 characterizes the expected utility–defense trade-off. When λ_{def} is small, MGSS preserves benign service accuracy more conservatively but provides weaker suppression of extractable supervision. As λ_{def} increases, the clone model receives less useful supervision and its accuracy is expected to decrease, but excessive shaping can begin to reduce the protected victim accuracy. The default setting $\lambda_{\text{def}} = 0.05$ is used as the balanced operating point in the main experiments because it substantially reduces clone accuracy while keeping the victim accuracy close to the undefended model.

Different model-extraction defenses use heterogeneous mechanisms, so an identical posterior-distortion control is not directly well-defined across methods. In the main comparisons, we match the dataset, attack protocol, query budget, victim and substitute architectures, and evaluation metrics whenever reproducible, while reporting extraction resistance alongside benign clean accuracy. The compared active defenses operate within a narrow clean-accuracy range on the reported datasets; the observed differences in extraction resistance therefore do not arise from obviously different sacrifices in clean utility. We interpret the main tables as practical protocol-matched comparisons at similar benign-task utility, rather than strict posterior-distortion-matched Pareto comparisons. Appendix B.2 characterizes MGSS's posterior-level utility, and this subsection characterizes its own defense–utility trade-off. We do not claim strict Pareto dominance under an identical posterior-distortion budget.

G.2 Coverage-Source Sensitivity

MGSS does not explicitly estimate the attacker's query distribution, but the diversity of defender-side coverage data can affect which maturity-discrepancy states are observed during offline training. To quantify this effect, we keep the remaining training and attack settings fixed and vary only the coverage source used by the defense branch in the CIFAR-10 DFME setting. These results are fixed-seed single-run diagnostics and are not intended to replace the five-run estimate reported in the main experiments. In particular, the default clone accuracy of 40.61% here is a single-run result, whereas $39.35 \pm 2.47\%$ in Table 1 is the five-run mean and standard deviation.

Table 13: Sensitivity to the defender-side coverage source on CIFAR-10 under DFME. These are fixed-seed supplementary diagnostics; lower clone accuracy and higher protected accuracy are better.

Defense-Branch Coverage Data	Protected Acc. (%)	DFME Clone Acc. (%)
CIFAR-10 strong only	94.36	52.13
CIFAR-10 strong + CIFAR-100 strong	94.39	45.28
CIFAR-100 strong only (default)	94.46	40.61

Protected-victim accuracy remains within 94.36%–94.46% across the tested coverage sources, while DFME clone accuracy varies from 40.61% to 52.13%. Thus, the coverage source materially affects defense strength, even though the protected utility remains similar. Importantly, MGSS does not require the defender-side coverage distribution to match the attacker's query distribution; rather, more effective or diverse coverage can expose a broader set of maturity-discrepancy states during offline training. Coverage diversity should therefore be viewed as a sensitivity factor of the defense rather than an explicit prior over the attacker's queries.

H Training Efficiency Evaluation

We compare the training cost of MGSS with representative extraction defenses on CIFAR-10. Unless otherwise specified, the reported training times are measured in our local environment with a single NVIDIA GeForce RTX 3090 GPU. MGSS freezes the victim backbone and trains only a lightweight residual head. This avoids full victim retraining and keeps the additional training stage short.

Table 14: Training time on CIFAR-10. Lower is better. Except for AAUG, all reported times are measured under our RTX 3090 environment.

Algorithm	Total Training Time (min)
MeCo	136.31
ACT	17.28
DNF	4.86
MGSS (Ours)	**3.69**

Table 14 shows that MGSS has substantially lower training cost than MeCo and ACT, and is slightly faster than DNF in our environment. Compared with defenses that require more extensive defensive training, MGSS provides a practical lightweight add-on to an already trained victim model.

AAUG is not included in Table 14 because, to the best of our knowledge, it does not provide an official public implementation, making it infeasible for us to reproduce its training time under the same RTX 3090 environment. We therefore report the training cost stated in the AAUG paper separately. AAUG reports two training protocols on a single NVIDIA A6000 GPU: full retraining takes 5.25 ± 0.17 hours, i.e., 315.00 ± 10.20 minutes, while fine-tuning takes 0.89 ± 0.10 hours, i.e., 53.40 ± 6.00 minutes. Since these numbers are reported under a different hardware and code-availability setting, we do not mix them with our local timing comparison.

I Parameter Overhead

We also report the trainable parameter overhead introduced by the MGSS residual head under the main experimental settings. The default victim is a ResNet34-8x model. The protected API keeps the victim backbone frozen and appends a defended residual head whose output is added to the original logits, i.e., $z_T(x) + g_\phi(z_T(x), M_T(x))$. Table 15 counts only trainable parameters. The prototype statistics used by the defended head are registered as buffers rather than trainable parameters, so they are not included in this count.

Table 15: Trainable parameter overhead of MGSS under the main experimental settings.

Dataset	ResNet34-8x Victim	MGSS Residual Head	Increase
CIFAR-10	21,282,122	307,562	1.445%
CIFAR-100	21,328,292	336,452	1.577%
MiniImageNet	21,328,292	336,452	1.577%

J Test-Time Memory Consumption Evaluation

We also evaluate memory consumption during inference. MGSS does not require test-time back-propagation, model ensembles, or randomized sampling. Its memory overhead mainly comes from computing the selected stage features and the lightweight residual head during the same deterministic forward path.

Table 16: Test-time memory consumption on CIFAR-10. Lower is better.

Algorithm	CIFAR-10
MeCo	2.03 GB
ACT	2.03 GB
AAUG	2.10 GB
DNF	1.74 GB
MGSS (Ours)	2.36 GB

Table 16 shows that MGSS has comparable test-time memory consumption to recent lightweight defenses.

K LIMITATIONS

MGSS has several limitations. First, our evaluation focuses primarily on image-classification model-extraction benchmarks. Extending maturity-guided supervision shaping to other modalities, structured prediction tasks, and large foundation-model APIs remains future work.

Second, MGSS assumes that the defender controls the deployed victim model and can access intermediate representations or stage-wise readouts. This assumption is natural for model owners deploying their own APIs, but it does not apply to settings where the defender can only access an externally hosted opaque black-box model.

Third, Appendix E.4 provides evidence that MGSS transfers to several non-default victim architectures. Nevertheless, the method still requires semantically meaningful ordered intermediate representations or readout locations. If an architecture does not provide such representations, or if its intermediate features do not yield stable class-support evidence, the maturity-discrepancy signal may become less reliable.

Fourth, MGSS is attacker-query-distribution agnostic in the sense that it does not estimate the attacker's query distribution, does not simulate a specific attacker strategy, and does not perform online attack-query detection. The offline data used to train the residual head play a calibration and probing role: task data anchor normal service utility, while coverage views expose diverse maturity-discrepancy states of the frozen victim. The shaping target for each coverage input is derived from the victim's own internal trajectory rather than from attacker labels, attacker queries, or an externally specified attack distribution. Therefore, the coverage views are not an attacker-query prior; they are used to teach the lightweight head how to map intrinsic maturity inconsistency to residual supervision shaping. Nevertheless, their diversity can affect how broadly the head observes high-discrepancy states during offline training. Appendix G.2 quantifies this sensitivity, showing that the tested coverage sources preserve similar protected-victim utility but yield different levels of extraction resistance.

Fifth, the shaping strength introduces a utility–defense trade-off. Stronger shaping can reduce the extractability of the exposed supervision, but it may also affect difficult, ambiguous, or distribution-shifted benign inputs. In practice, the defense strength should be selected under a validation utility constraint. Appendix B.2 further evaluates posterior fidelity and calibration on CIFAR-10 and CIFAR-100, complementing clean top-1 accuracy as a service-utility measure. Broader evaluation on probability-sensitive downstream applications and distribution-shifted benign inputs remains future work.

Finally, our evaluation focuses on data-free extraction attacks. Section 5.4 additionally evaluates adaptive DFME variants, with their attack details provided in Appendix E.3. The tested adaptive variants partially recover clone accuracy and do not establish immunity to adaptive extraction. Exhaustive evaluation against a broader range of adaptive recovery strategies remains future work.