Dynamic Neural Fortresses: An Adaptive Shield for Model Extraction Defense

Siyu Luan[1*], Zhenyi Wang[2#*], Li Shen[3], Zonghua Gu[4], Chao Wu[5], Dacheng Tao[6]

[1]University of Copenhagen, Denmark [2]University of Maryland, College Park, USA
[3] Shenzhen Campus of Sun Yat-sen University [4]Hofstra University, USA
[5]University at Buffalo, USA [6]Nanyang Technological University, Singapore
Siyu Luan and Zhenyi Wang contribute equally. Corresponding author: Zhenyi Wang

Introduction

Model extraction aims to acquire a pre-trained black-box model concealed behind a black-box API. Existing defense strategies against model extraction primarily concentrate on preventing the unauthorized extraction of API functionality. However, two significant challenges still need to be solved:

- Neural network architecture also requires protection.
- Using the same network architecture for both attack and benign queries results in substantial resource wastage.

To address these challenges, we propose Dynamic Neural Fortresses (DNF), employing a dynamic Early-Exit Neural Network (EENN), deviating from the conventional fixed architecture.

Dynamic Early-Exit Strategy

DNF uses an Early-Exit Neural Network (EENN) to strengthen defenses against model extraction attacks.

- **Attack queries** are encouraged to exit randomly at early layers, which: (1) Outputs non-semantic, misleading information to attackers. (2) Reduces computational cost and improves efficiency. (3) Increases difficulty for attackers to infer the model architecture.
- **Benign queries** are guided to exit at later layers, ensuring they receive meaningful and accurate outputs.

Learning Objective

To implement our strategy, we draw inspiration from the deep information bottleneck theory.

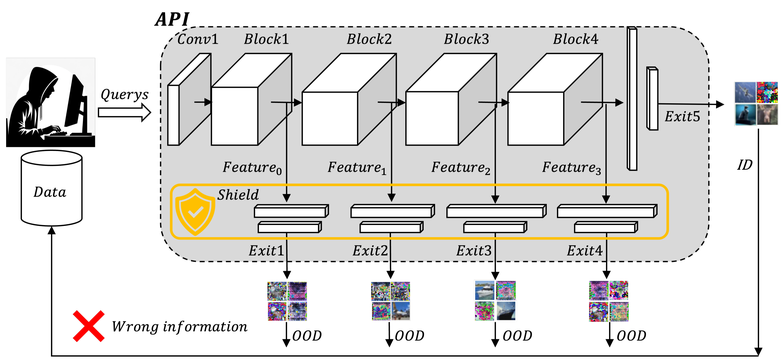
For **out-of-distribution (OOD) data**, the goal is to degrade prediction quality for defense:

- Minimize mutual information between latent features and labels to weaken prediction ability.
- Maximize mutual information between input and latent features to focus on non-semantic information, leading to inaccurate predictions.

For **in-distribution (ID) data**, the goal is to preserve model utility::

- Maximize mutual information between latent features and labels to strengthen predictive performance.
- Minimize mutual information between input and latent features to extract compressive and semantic content for better utility.

Illustration of DNF



ID data is expected to exit at the later layers of the EENN, after more feature extraction operations. Conversely, OOD data (attack queries) is anticipated to exit non-deterministically at earlier layers, undergoing fewer feature extraction operations.

The formula for Learning Objective

To obfuscate information gathered by potential attackers, our goal is to maximize the disparity between exit layers for OOD and ID data. This is achieved by introducing an entropy-based regularization term $H(V_i^*(x))$ at each exit layer, which measures the uncertainty of the model's prediction. The entropy is computed from the victim model's output h at the i^{th} exit, where J is the number of classes, as shown in the following equation:

$$H(h) = -\sum_{j=1}^{j=J} V_{i,j}^*(x, \delta_V^i) \log V_{i,j}^*(x, \delta_V^i) \tag{1}$$

ID Data Learning Objective We present the following learning objectives for ID queries.

$$L_{id} = \sum_{i=1}^{i=N} [I(X_{id}; Z_i) - I(Z_i; Y_{id}) + \alpha_i K(i) H(V_i^*(x_{id}))] \tag{2}$$

OOD Data Learning Objective Conversely, we put forth the following objectives for OOD queries:

$$L_{ood} = \sum_{i=1}^{i=N} [I(Z_i; Y_{ood}) - I(X_{ood}; Z_i) + \alpha_i J(i) H(V_i^*(x_{ood}))] \tag{3}$$

The overall learning objective is to minimize the following loss:

$$L_d = L_{id} + L_{ood} \tag{4}$$

Defense against Data-free Model Extraction

The results of defense against soft-label and hard-label DFME attack on CIFAR10 and CIFAR100 are shown in Table 1. Our EENN's backbone is based on ResNet34. DNF (our method) outperforms the baselines in terms of clone model accuracy.

Table 1. Clone model accuracy on **CIFAR-10** and **CIFAR-100** with *ResNet34* as the victim model.

Attack	Defense	CIFAR-10 Clone Model Architecture			CIFAR-100 Clone Model Architecture		
		ResNet18-8X	MobileNetV2	DenseNet121	ResNet18-8X	MobileNetV2	DenseNet121
DFME (Soft-label)	Undefended ↓	87.36 ± 0.78%	75.23 ± 1.53%	73.89 ± 1.29%	58.72 ± 2.82%	28.36 ± 1.97%	27.28 ± 2.08%
	RandP ↓	84.28 ± 1.37%	70.56 ± 2.23%	70.03 ± 2.38%	41.69 ± 2.91%	22.75 ± 2.19%	23.61 ± 2.70%
	P-poison ↓	78.06 ± 1.73%	66.32 ± 1.36%	68.75 ± 1.40%	38.72 ± 3.06%	20.87 ± 2.61%	21.89 ± 2.93%
	GRAD ↓	79.33 ± 1.68%	65.82 ± 1.67%	69.06 ± 1.57%	39.07 ± 2.72%	20.71 ± 2.80%	22.08 ± 2.78%
	MeCo ↓	**51.68 ± 1.96%**	46.53 ± 2.09%	61.38 ± 2.41%	29.57 ± 1.97%	12.18 ± 1.05%	10.79 ± 1.36%
	DNF ↓	53.91 ± 2.30%	**46.32 ± 1.45%**	**47.21 ± 2.15%**	**18.03 ± 3.03%**	**10.82 ± 1.34%**	**6.75 ± 1.23%**
DFMS-HL (Hard-label)	Undefended ↓	84.67 ± 1.90%	79.28 ± 1.87%	68.87 ± 2.08%	72.57 ± 1.28%	62.71 ± 1.68%	63.58 ± 1.79%
	RandP ↓	84.02 ± 2.31%	78.71 ± 1.93%	68.16 ± 2.23%	72.43 ± 1.43%	62.06 ± 1.82%	63.16 ± 1.73%
	P-poison ↓	84.06 ± 1.87%	79.02 ± 1.96%	68.05 ± 2.17%	71.83 ± 1.32%	61.83 ± 1.79%	62.73 ± 1.91%
	GRAD ↓	84.28 ± 1.95%	78.83 ± 1.91%	68.11 ± 1.93%	71.89 ± 1.37%	62.60 ± 1.71%	62.57 ± 1.80%
	MeCo ↓	76.86 ± 2.09%	**71.22 ± 1.87%**	62.33 ± 2.01%	59.30 ± 1.70%	55.32 ± 1.65%	56.80 ± 1.86%
	DNF ↓	**76.51 ± 2.12%**	75.01 ± 1.25%	**61.02 ± 1.21%**	**52.98 ± 2.24%**	**48.41 ± 1.78%**	**49.72 ± 1.24%**

Defense against Data-based Model Extraction

To verify the effectiveness of DNF on CUB200 and ImageNet200 datasets, our EENN's backbone is ResNet50 and Swin Transformer, respectively. The results are shown in Table 2. DNF (our method) outperforms the baselines in terms of clone model accuracy.

Table 2. Clone model accuracy on **CUB200** and **ImageNet200** with *ResNet50* as the victim model.

Attack	Defense	CUB200 Clone Model Architecture			ImageNet200 Clone Model Architecture			
		ResNet50	ResNet34	VGG19	ViT-Large	CaiT	DeiT	Swin-Large
Soft-label	Undefended ↓	55.49 ± 1.50%	36.71 ± 0.79%	39.85 ± 1.71%	73.35 ± 0.89%	62.26 ± 1.24%	56.59 ± 1.54%	60.67 ± 1.43%
	RandP	39.77 ± 0.35%	20.59 ± 1.03%	24.02 ± 0.35%	69.69 ± 1.08%	59.58 ± 1.78%	53.97 ± 1.89%	56.83 ± 0.98%
	P-poison	24.94 ± 1.25%	15.31 ± 1.17%	20.81 ± 1.33%	65.53 ± 1.16%	58.91 ± 1.74%	52.08 ± 0.53%	55.67 ± 1.58%
	GRAD	24.32 ± 0.55%	15.06 ± 1.11%	20.65 ± 0.24%	65.32 ± 0.68%	59.24 ± 0.75%	51.97 ± 1.82%	55.79 ± 0.57%
	MeCo	51.32 ± 0.54%	31.98 ± 0.73%	34.36 ± 0.25%	69.93 ± 0.32%	60.17 ± 0.20%	53.78 ± 0.79%	58.85 ± 0.73%
	DNF ↓	**15.32 ± 1.21%**	**5.27 ± 1.72%**	**8.21 ± 2.23%**	**56.03 ± 1.12%**	**46.21 ± 0.91%**	**45.31 ± 1.09%**	**42.10 ± 1.26%**
Hard-label	Undefended ↓	31.29 ± 1.58%	21.57 ± 0.62%	23.27 ± 0.80%	63.57 ± 0.69%	57.73 ± 0.80%	53.16 ± 1.52%	60.26 ± 1.23%
	RandP	30.89 ± 0.61%	21.68 ± 0.91%	23.44 ± 1.34%	63.18 ± 1.14%	57.31 ± 0.80%	52.86 ± 0.45%	59.58 ± 1.01%
	P-poison	30.69 ± 0.91%	21.54 ± 1.98%	22.06 ± 1.01%	63.09 ± 0.65%	57.12 ± 0.51%	52.57 ± 1.12%	59.23 ± 1.87%
	GRAD	31.23 ± 0.61%	22.38 ± 1.52%	22.37 ± 0.44%	63.21 ± 1.24%	57.23 ± 1.90%	52.34 ± 1.54%	59.30 ± 1.23%
	MeCo	29.42 ± 0.46%	19.90 ± 0.42%	21.08 ± 0.34%	63.31 ± 0.48%	57.25 ± 0.24%	52.69 ± 0.45%	59.61 ± 0.50%
	DNF ↓	**16.01 ± 1.24%**	**10.91 ± 1.02%**	**12.98 ± 1.25%**	**48.95 ± 2.11%**	**43.65 ± 0.91%**	**40.98 ± 1.69%**	**44.01 ± 0.82%**

Key Contributions

We present the first defense framework that provides three key protective benefits simultaneously:

1. Safeguarding the model's functionality while substantially lowering the accuracy of cloned models.
2. Improving computational efficiency by dynamically allocating resources to queries.
3. Securing the model architecture from extraction.