A. Additional Details

A.1. Visualization of the Unlearnable Examples

Figure 3 provides a visualization of unlearnable examples synthesized using the UE methods on an example image from the Oxford Flowers-102 dataset.

A.2. Augmentation Strategies

A^3 includes augmentation strategies for both the image and text modalities.

For the image modality, the augmentation pipeline applies the following strategies in sequence: PlasmaContrast and PlasmaBrightness from Tormentor [21], which increases diversity of augmented samples while preserving important features, channel shuffling, which randomly permutes the color channels, and TrivialAugment [20], which is known to improve model training generalization. Figure 4 shows the visualizations of the individual image augmentation strategies.

The text modality augmentation strategies primarily involve adding a small uniform noise, masking random tokens, randomly flipping embedding vectors, as well as random rotation with a small angle. These augmentation operations are applied only during training to help improve the model's robustness against UEs.

The specific configurations for both modalities are shown in Tab. 5.

A.3. The overall training algorithm for A^3

B. Experimental Setup

B.1. Datasets

We evaluated A^3 on 7 datasets, which are widely used in the literature related to prompt learning. Below, we will briefly summarize these datasets, and provide example images of some datasets in Figure 5.

- The **ImageNet** dataset [6] is a large-scale visual database containing over 1.2 million labeled images across 1000 categories. It is widely used for image classification and object detection tasks.
- The **Caltech-101** dataset [8] is an image classification dataset provided by the California Institute of Technology. It consists of 101 object categories, with each category containing around 40 to 800 images. It is commonly used for object recognition and image classification research.
- The **Oxford Pets** dataset [23] contains images of 37 pet species, with over 7,349 images. The dataset is aimed at pet species recognition, and the images vary in terms of pose, background, and lighting conditions.
- The **Oxford Flowers** dataset [22] consists of 102 flower categories, with each category containing around 40 to 258 images. It is used for flower classification tasks, and the images often have varied backgrounds and lighting, making it a challenging dataset.
- The **Food-101** dataset [3] contains 101 food categories, each with about 1,000 images. It is designed for food classification research, featuring a wide variety of food images such as pizza, burgers, sushi, *etc*.
- The **SUN-397** dataset [37] is a large-scale dataset designed for scene classification tasks. It contains 397 scene categories, covering a wide range of environments such as beaches, forests, highways, and urban settings. The dataset includes over 100,000 images in total, with approximately 200 images per category.
- The **UCF-101** dataset [31] is a video dataset containing 101 action categories, with approximately 100 video clips per category. It is commonly used for action recognition tasks and includes a wide range of activities, such as running, swimming, dancing, *etc*.

B.2. Methods for Learning from UEs

In our experiments, we compared several existing learning strategies, such as **Gray**scale, **JPEG** compression, adversarial training (**AT**), and **UEraser**. This sections will provide a detailed explanation of the methods and parameter settings.

- **Adversarial Training** (AT) [19] is a method that generates adversarial examples during training and use them to train the model, and it is traditionally shown to improve model robustness against adversarial attacks. By exposing the model to adversarial perturbations during training, the trained model may be less sensitive to small perturbations in the input data. It is also shown to be effective to learn from UEs. Our AT uses a PGD-based adversarial training algorithm on top of the standard CoCoOp training algorithm, where the perturbation budget is $8/255$ under the ℓ_∞ norm, the PGD step size is $2/255$, step count is 7, and the number of training epochs is 20.
- **Gray**scale conversion, from [16], is a simple image processing where only intensity (brightness) information is retained, and color details are discarded. As many UEs form strong shortcuts based on color information, it helps by simplifying the input images and reducing the impact of color-based adversarial manipulations. In our experiments, we converted the images to grayscale, and used the standard CoCoOp training algorithm with 20 epochs.
- **JPEG** is a widely-used lossy image compression format which typically introduces blocking and blurry compression artifacts. Based on the fact that perturbations are often high-frequency and fine-grained, it may be an effective way to disrupt perturbations from UEs. Applying JPEG compression helps by degrading the quality of UEs, making perturbations less effective, while the underlying content remains mostly intact. We follow [16] and used a JPEG compression rate of 10, but adapted it to the standard CoCoOp prompt learning.

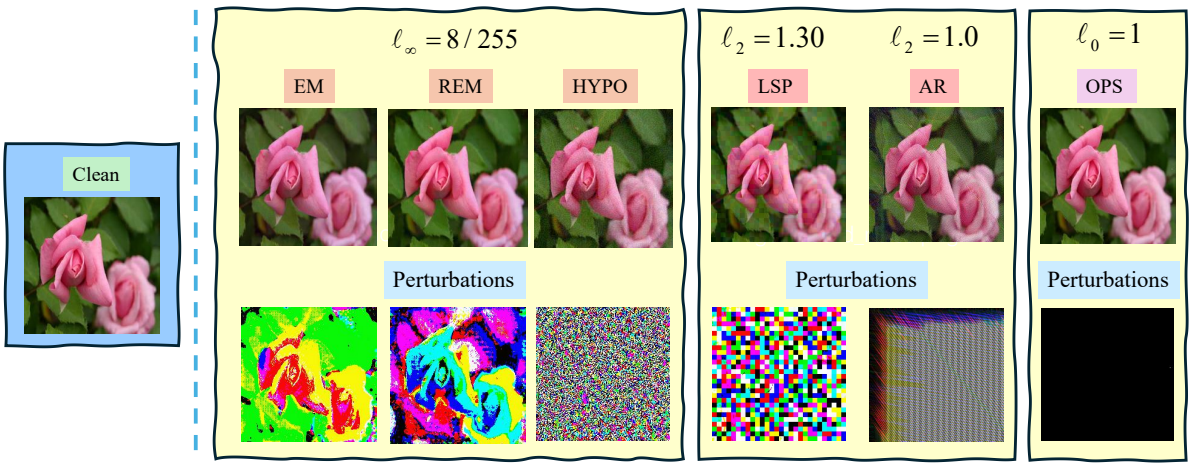

Figure 3. The visualization of unlearnable examples and their respective perturbations on an example image from the Oxford Flowers-102 dataset.

Table 5. Augmentation strategies for text and image modalities of A^3.

Modality	Augmentation Strategy	Probability	Configuration
Text	Uniform Noise	0.2	Noise $\sim \mathcal{U}(-0.05, 0.05)$
	Mask	1.0	Mask rate = 0.2
	Flip	0.2	—
	Rotation	0.2	Angle $\sim \mathcal{U}(-10°, 10°)$
Image	Plasma Brightness	1.0	Roughness = (0.3, 0.7), Intensity = (0.5, 1.0)
	Plasma Contrast	1.0	Roughness = (0.3, 0.7)
	Channel Shuffle	0.5	—
	Trivial Augment	1.0	—

Figure 4. Visualization for the image augmentation strategies in A^3 for an example Caltech-101 image.

- **UEraser** employs a sequence of image augmentations and loss-maximizing augmentations. We use the standard hyperparameters from the official implementation [25], and integrate it with the CoCoOp training algorithm by applying UEraser augmentations to the images before feeding them to the image encoder.

C. Additional Results

C.1. Using other PL algorithms for A^3

In Table 6, we have conducted further experiments extending other PL methods with A^3, following the ablation analyses in Table 3. Notably, the results of adopting other PL methods are less competitive than those in Table 3, justifying our choice of CoCoOp for A^3.

Table 6. Prompt learning methods. (Caltech-101, RN-50, 16-shot)

Methods	EM $^8/_{255}$	REM $^8/_{255}$	HYPO $^8/_{255}$	LSP 1.30	AR 1.00	OPS 1
CoOp	70.10	50.63	44.81	41.22	47.37	85.45
+Text	80.04	81.62	79.69	79.75	80.89	88.50
+Image	89.57	89.31	88.60	88.32	90.15	92.48
+Full	93.95	93.47	92.89	93.60	92.28	93.04
ProDA	72.68	51.16	46.29	44.05	50.77	86.03
+Text	81.56	82.13	80.48	80.87	81.06	89.86
+Image	91.89	90.67	90.03	89.74	90.26	92.71
+Full	94.16	94.00	93.42	94.05	93.77	93.65
KgCoOp	68.33	45.82	40.26	38.53	44.96	83.47
+Text	79.68	80.09	79.01	79.20	80.16	87.42
+Image	88.91	89.14	88.00	88.48	89.75	91.07
+Full	93.23	92.98	92.15	91.93	92.16	92.82

Algorithm 1 The overall training algorithm for A^3.

function A^3(training set $\mathcal{D}_{\text{ue}}$, image augmentation policies $\mathcal{A}_{\text{im}}$, text augmentation policies $\mathcal{A}_{\text{tx}}$, image encoder f_{im}, text encoder f_{tx}, meta-net m, learning rate α, batch size B, number of training iterations N, number of image augmentations K_{im}, number of text augmentations K_{tx}, similarity function sim, all trainable weights $\boldsymbol{\phi} = [\text{meta-net weights } \boldsymbol{\psi}, \text{prompt embeddings } \mathbf{V}]$)
  for $n \in [1, \ldots, N]$ **do**
    $\mathbf{B} \leftarrow \text{mini-batch}_B(\mathcal{D}_{ue})$ ▷ Sample a mini-batch from $\mathcal{D}_{\text{ue}}$
    for $\mathbf{x}^{(m)}, y^{(m)} \in \mathbf{B}$ **do** ▷ For each image in mini-batch...
      for $i \in [1, \ldots, K_{\text{im}}]$ **do**
        $\tilde{\mathbf{x}}_i^{(m)} \leftarrow a_{\text{im}}(\mathbf{x}_i)$, where $a_{\text{im}} \sim \mathcal{A}_{\text{im}}$ ▷ Sample K_{im} image augmentations
        for $j \in [1, \ldots, K_{\text{tx}}]$ **do**
          $\tilde{\mathbf{t}}_j^{(m)} \leftarrow a_{\text{tx}}([\mathbf{v}_{k \bmod M}, \mathbf{c}_{y_i}])$, where $a_{\text{tx}} \sim \mathcal{A}_{\text{tx}}$ ▷ Sample K_{tx} text augmentations ...
          $\tilde{\mathbf{t}}_j^{(m)} \leftarrow \tilde{\mathbf{t}}_j^{(m)} + m_{\boldsymbol{\psi}}(\tilde{\mathbf{x}}_k^{(m)})$, where $k \sim \mathcal{U}\{1, K_{\text{im}}\}$ ▷ ...and apply meta-net augmentations
          $\mathcal{S}_{ij}^{(m)} \leftarrow \text{sim}(f_{\text{im}}(\tilde{\mathbf{x}}_i^{(m)}), f_{\text{tx}}(\tilde{\mathbf{t}}_j^{(m)}))$ ▷ Compute similarities between the augmented image-text pairs
        end for
      end for
      $\ell(\mathbf{x}^{(m)}, y^{(m)}) \leftarrow \mathcal{L}(\min_{i,j} \mathcal{S}_{ij}^{(m)}, y_i)$ ▷ Find the least similar pair, and compute its alignment loss
    end for
    $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \alpha \nabla_\theta \frac{1}{B} \sum_{k=1}^{B} \ell(\mathbf{x}^{(m)}, y^{(m)})$ ▷ SGD on the mini-batch $\mathbf{B}$ of the max-loss image-text pairs
  end for
  return $\boldsymbol{\phi}$ ▷ Return the learned prompt embeddings and meta-net weights
end function

(a) Oxford Flowers-102.

(b) Oxford Pets.

(c) Caltech-101.

Figure 5. Example images from the datasets used in our experiments.

C.2. Transferring UEs across models and PL algorithms

In Table 7, we explore the transferability of EM UEs. Here, we take the UEs generated by ViT-B/16, using either CoOp or CoCoOp as the PL algorithm, and learn them using the other algorithm on different architectures. The result shows that UEs can harm PL, and degrade accuracy significantly from the zero-shot case ($\alpha_{\text{b}} = 96.30\%, \alpha_{\text{n}} = 93.23\%, \alpha_{\text{h}} = 94.74\%$), and are transferable.

Table 7. Transfer EM UEs across models & PL methods on Caltech-101.

ViT-B/16 →		ViT-B/16	RN-50	RN-101	ViT-B/32
CoCoOp	α_b	79.43	69.82	75.35	83.78
↓	α_n	73.09	63.48	69.17	78.64
CoOp	α_h	76.18	66.53	71.67	81.10
CoOp	α_b	82.12	72.53	75.56	86.34
↓	α_n	76.95	66.95	70.08	80.79
CoCoOp	α_h	79.55	69.54	72.73	83.47
CoCoOp	α_b	80.68	70.05	75.93	82.76
↓	α_n	74.90	63.97	69.60	79.57
CoCoOp	α_h	77.54	66.88	72.58	81.06

C.3. A^3-Adapted UEs

In this section, we consider an adaptive-variant of EM, where the content creator is aware of the use of A^3 by the learner. For this, it adapts Objective 9 of A^3 to EM to generate perturbations $\boldsymbol{\delta}$:

$$\min_{(\boldsymbol{\delta}, \boldsymbol{\phi})} \mathbb{E}_{(\mathbf{x}_l, y_l) \sim \mathcal{D}_{\text{ue}}} \Big[\max_{(i,j)} \mathcal{L}(\mathcal{S}(\tilde{\mathbf{x}}_l + \boldsymbol{\delta}_l, \tilde{\mathbf{t}})_{ij}, y) \Big], \tag{11}$$

In addition, we also examine the ablation of image- and text-based augmentations for both the content creator and the learner. We provide the results in Table 8.

From Table 8, it can be observed that the A^3 does not significantly reduce its defense effectiveness when faced with A^3-adapted UEs. We believe the main reason is that while UEs are bounded by the perturbation budget, A^3's transformations are not, and can thus be effective in reducing the impact of UEs.

Table 8. Adaptive poisoning of A^3 variants with EM on Caltech-101. Rows indicate the methods to generate the poisoning samples. and columns indicate the augmentation modalities to train the prompt learner. "+ Image" and "+ Text" respectively denote using only the image and text augmentation modalities, and "Full" denotes using both. The image encoder backbone is ResNet-50.

Methods	**Baseline**	**+Image**	**+Text**	**Full**
EM	75.53	92.51	84.69	94.28
+ Image	78.10	87.84	81.07	90.66
+ Text	76.29	90.13	82.94	92.75
Full	79.68	86.37	80.83	89.42

Table 9. A^3 with varying K_{im} and a fixed $K_{\text{tx}} = 5$, and *vice versa* for a varying K_{tx} and a fixed $K_{\text{im}} = 5$, under EM on Caltech-101 with the base-to-novel protocol. The image encoder backbone is ResNet-50.

K_{tx}	α_{b}	α_{n}	α_{h}
1	93.00	90.54	91.75
3	94.11	91.08	92.57
5	94.43	91.22	92.80
7	93.75	90.61	92.15
9	94.49	91.35	92.89

K_{im}	α_{b}	α_{n}	α_{h}
1	86.82	83.46	85.11
3	91.50	88.12	89.78
5	94.39	91.24	92.79
7	94.74	90.58	92.61
9	94.96	91.63	93.27

C.4. Sensitivity Analyses

In this section, to evaluate the A^3 prompt learning under different numbers of repeated augmentation samples K_{im} and K_{tx}, We first fix K_{im} to 5 and sweep the results of $K_{\text{tx}} \in \{1, 3, 5, 7, 9\}$, and *vice versa* for a K_{im} sweep. We present the results for the base-to-novel protocol, which reports the accuracy values of the base (α_{b}) and novel (α_{n}) classes, and the harmonic mean (α_{h}).

Table 9 present the results that respectively sweep the numbers of image and text augmentation samples. As an augmented sample corresponds to an additional pass through the image or text encoder, the computational cost increases linearly with K_{im} or K_{tx}. In both cases, as K_{im} or K_{tx} increases, the performance improves, but mostly saturates after a certain point. For this reason, we recommend $K_{\text{im}} = K_{\text{tx}} = 5$ to strike a balance between performance and computational cost.