Supplementary Document
for
MRCert: Towards Post-deployment Patch Robustness Certification for Adversarially Patched Samples via Type-specific Masking

A Formal Proof

Property (Infeasibility of Re-certifying Adversarially Patched Samples for Existing Masking-based Recovery Works)**.** $\forall x' \in \mathbb{A}_{\mathbb{P}}(x), c_r(x) \wedge f(x') \neq f(x) \implies \neg c_r(x')$.

Proof. We know in [Xiang *et al.*, 2022], $c_r(\hat{x}) := [\forall \mathrm{M}_1, \mathrm{M}_2 \in \mathbb{M}, f(\hat{x} \odot \mathrm{M}_1 \odot \mathrm{M}_2) = f(\hat{x})]$. Then if $c_r(x) =$ *True*, we know $[\forall \mathrm{M}_1, \mathrm{M}_2 \in \mathbb{M}, f(x \odot \mathrm{M}_1 \odot \mathrm{M}_2) = f(x)]$. We also know $\forall \mathrm{P} \in \mathbb{P}, \exists \mathrm{M} \in \mathbb{M}_{\mathbb{P}}, \mathrm{M} \odot \mathrm{P} = \mathrm{P}$ by definition, which makes $\forall x' \in \mathbb{A}_{\mathbb{P}}(x), x' \odot \mathrm{M} = x \odot \mathrm{M}$ (see Fig. 2). Therefore, $\exists x' \in \mathbb{A}_{\mathbb{P}}(x), f(x' \odot \mathrm{M}) = f(x \odot \mathrm{M}) = f(x)$. We have $f(x') \neq f(x)$ from the antecedent; therefore, $\exists x' \in \mathbb{A}_{\mathbb{P}}(x), f(x' \odot \mathrm{M}) \neq f(x')$. Since $c_r(x')$ requires $[\forall \mathrm{M}_1, \mathrm{M}_2 \in \mathbb{M}, f(x' \odot \mathrm{M}_1 \odot \mathrm{M}_2) = f(x')]$, $c_r(x') =$ *False* holds. □

Lemma (Certification of benign samples)**.** *Given an arbitrary sample $\hat{x}$, if $[\forall \mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_3 \in \mathbb{M}, f(\hat{x} \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3) = f(\hat{x})]$ holds, $\forall \hat{x}' \in \mathbb{A}_{\mathbb{P}}(\hat{x}), g(\hat{x}') = g(\hat{x})$.*

Proof. We first analyze $g(\hat{x})$. Since $[\forall \mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_3 \in \mathbb{M}, f(\hat{x} \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3) = f(\hat{x})] \implies [\forall \mathrm{M}_1, \mathrm{M}_2 \in \mathbb{M}, f(\hat{x} \odot \mathrm{M}_1 \odot \mathrm{M}_2) = f(\hat{x})]$, we know its prediction label $g(\hat{x}) = f(\hat{x})$ output in Case ①. We then analyze $g(\hat{x}')$. For $\forall \hat{x}' \in \{\hat{x}' \mid \hat{x}' = (\mathrm{J} - \mathrm{P}) \odot \hat{x} + \mathrm{P} \odot \hat{x}' \wedge \mathrm{P} \in \mathbb{P}\}$ (i.e., $\forall \hat{x}' \in \mathbb{A}_{\mathbb{P}}(\hat{x})$), w.l.o.g., we let $\mathrm{M}_1 \odot \mathrm{P} = \mathrm{O}$. Therefore, we get $\hat{x}' \odot \mathrm{M}_1 = ((\mathrm{J} - \mathrm{P}) \odot \hat{x} + \mathrm{P} \odot \hat{x}') \odot \mathrm{M}_1 = \hat{x} \odot \mathrm{M}_1$. Then, we get $[\exists \mathrm{M}_1 \in \mathbb{M}, \forall \mathrm{M}_2, \mathrm{M}_3 \in \mathbb{M}, f(\hat{x}' \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3) = f(\hat{x})]$ (see Fig. 2). Note that the special cases $\mathrm{M}_2 = \mathrm{M}_3$, $\mathrm{M}_1 = \mathrm{M}_2 = \mathrm{M}_3$ are included. Case 1: Suppose the returned label of $\hat{x}'$ output in Case ① as $f(\hat{x}')$ (i.e., $\forall \mathrm{M}_1, \mathrm{M}_2 \in \mathbb{M}, f(\hat{x}' \odot \mathrm{M}_1 \odot \mathrm{M}_2) = f(\hat{x}')$), then we know $g(\hat{x}') = f(\hat{x}') = f(\hat{x}) = g(\hat{x})$. Case 2: Otherwise, its returned label should be output in Case ② as $f(\hat{x}' \odot \mathrm{M}_1)$, since $[\exists \mathrm{M}_1 \in \mathbb{M}, \forall \mathrm{M}_2, \mathrm{M}_3 \in \mathbb{M}, f(\hat{x}' \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3) = f(\hat{x}' \odot \mathrm{M}_1) = f(\hat{x})]$, and further $g(\hat{x}') = f(\hat{x}' \odot \mathrm{M}_1) = f(\hat{x}) = g(\hat{x})$. □

Lemma (Certification of adversarially patched samples)**.** *Given an arbitrary sample $\hat{x}$, if $[\exists \mathrm{M}_1 \in \mathbb{M}, \forall \mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4 \in \mathbb{M}, f(\hat{x} \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3 \odot \mathrm{M}_4) = f(\hat{x} \odot \mathrm{M}_1)]$ holds, $\forall \hat{x}' \in \mathbb{A}_{\mathbb{P}}(\hat{x}), g(\hat{x}') = g(\hat{x})$.*

Proof. We first analyze $g(\hat{x})$. Case 1: If $\hat{x}$ meet the condition of Case ①, we can further get $\forall \mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4 \in \mathbb{M}, f(\hat{x} \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3 \odot \mathrm{M}_4) = f(\hat{x})$. Note that the special case $\mathrm{M}_1 = \mathrm{M}_2 = \mathrm{M}_3 = \mathrm{M}_4$ is included, which means we can get $g(\hat{x}) = f(\hat{x} \odot \mathrm{M}_1)$. Case 2: Otherwise, by the given condition $[\exists \mathrm{M}_1 \in \mathbb{M}, \forall \mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4 \in \mathbb{M}, f(\hat{x} \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3 \odot \mathrm{M}_4) = f(\hat{x} \odot \mathrm{M}_1)] \implies [\exists \mathrm{M}_1 \in \mathbb{M}, \forall \mathrm{M}_2, \mathrm{M}_3 \in \mathbb{M}, f(\hat{x} \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3) = f(\hat{x} \odot \mathrm{M}_1)]$, $\hat{x}$ meet the condition in Case ② and its prediction label $g(\hat{x}) = f(\hat{x} \odot \mathrm{M}_1)$, same as Case ①. We then analyze $g(\hat{x}')$. For $\hat{x}' \in \{\hat{x}' \mid \hat{x}' = (\mathrm{J} - \mathrm{P}) \odot \hat{x} + \mathrm{P} \odot \hat{x}' \wedge \mathrm{P} \in \mathbb{P}\}$ (i.e., $\forall \hat{x}' \in \mathbb{A}_{\mathbb{P}}(\hat{x})$), Case 1: Suppose $\mathrm{M}_1 \odot \mathrm{P} = \mathrm{O}$. Then we can get $\hat{x}' \odot \mathrm{M}_1 = ((\mathrm{J} - \mathrm{P}) \odot \hat{x} + \mathrm{P} \odot \hat{x}') \odot \mathrm{M}_1 = \hat{x} \odot \mathrm{M}_1$, and further get $[\forall \mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4 \in \mathbb{M}, f(\hat{x}' \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3 \odot \mathrm{M}_4) = f(\hat{x} \odot \mathrm{M}_1)]$, which is the same condition as that on $\hat{x}$. Therefore, repeating those analysis for $g(\hat{x})$ above can get $g(\hat{x}) = g(\hat{x}')$. Case 2: Otherwise, for $\mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4$, w.l.o.g., we let $\mathrm{M}_2 \odot \mathrm{P} = \mathrm{O}$ ($\mathrm{M}_1 \neq \mathrm{M}_2$). Then similarly, we get $[\exists \mathrm{M}_1, \mathrm{M}_2 (\neq \mathrm{M}_1) \in \mathbb{M}, \forall \mathrm{M}_3, \mathrm{M}_4 \in \mathbb{M}, f(\hat{x}' \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3 \odot \mathrm{M}_4) = f(\hat{x} \odot \mathrm{M}_1)]$. Note that the special cases $\mathrm{M}_1 = \mathrm{M}_3, \mathrm{M}_2 = \mathrm{M}_4, \mathrm{M}_3 = \mathrm{M}_4$ are included. Case 2.1: Suppose the prediction label $g(\hat{x}')$ output in Case ①. Then, since $[\exists \mathrm{M}_1, \mathrm{M}_2 (\neq \mathrm{M}_1) \in \mathbb{M}, f(\hat{x}' \odot \mathrm{M}_1 \odot \mathrm{M}_2) = f(\hat{x} \odot \mathrm{M}_1)]$ (special case $\mathrm{M}_1 = \mathrm{M}_3$, $\mathrm{M}_2 = \mathrm{M}_4$), by the condition of Case ①, we know $g(\hat{x}') = f(\hat{x}') = f(\hat{x} \odot \mathrm{M}_1) = g(\hat{x})$. Case 2.2: Suppose the prediction label $g(\hat{x}')$ output in Case ②. Then, since $[\exists \mathrm{M}_1, \mathrm{M}_2 (\neq \mathrm{M}_1) \in \mathbb{M}, \forall \mathrm{M}_3 \in \mathbb{M}, f(\hat{x}' \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3) = f(\hat{x} \odot \mathrm{M}_1)]$ (special case $\mathrm{M}_3 = \mathrm{M}_4$), by the condition of Case ②, we know $g(\hat{x}') = f(\hat{x}' \odot \mathrm{M}_1) = f(\hat{x} \odot \mathrm{M}_1) = g(\hat{x})$. Case 2.3: Otherwise, the prediction label of $\hat{x}'$ should output in Case ③ since $[\exists \mathrm{M}_1, \mathrm{M}_2 (\neq \mathrm{M}_1) \in \mathbb{M}, \forall \mathrm{M}_3, \mathrm{M}_4 \in \mathbb{M}, f(\hat{x}' \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3 \odot \mathrm{M}_4) = f(\hat{x} \odot \mathrm{M}_1) = f(\hat{x}' \odot \mathrm{M}_1 \odot \mathrm{M}_2)]$ (special case $\mathrm{M}_1 = \mathrm{M}_3$, $\mathrm{M}_2 = \mathrm{M}_4$), and further $g(\hat{x}') = f(\hat{x}' \odot \mathrm{M}_1 \odot \mathrm{M}_2) = f(\hat{x} \odot \mathrm{M}_1) = g(\hat{x})$. □

Theorem (Certification of samples)**.** *Given an arbitrary sample $\hat{x}$, if $c(\hat{x}) = True$ holds, $\forall \hat{x}' \in \mathbb{A}_{\mathbb{P}}(\hat{x}), g(\hat{x}') = g(\hat{x})$.*

Simply conjoining the antecedent of Lemma 1 and Lemma 2 can prove this theorem.

Theorem (Round-trip certification of samples)**.** *Given a benign sample x, if $[\forall \mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4 \in \mathbb{M}, f(x \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3 \odot \mathrm{M}_4) = f(x)]$ (i.e., $c_r^2(x) =$ True), $[\forall x' \in \mathbb{A}_{\mathbb{P}}(x), g(x') = g(x) \wedge c_r(x') =$ True $\wedge\, c_r(x) =$ True$]$.*

Proof. By the condition $[\forall \mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4 \in \mathbb{M}, f(x \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3 \odot \mathrm{M}_4) = f(x)]$, we know the returned label $g(x) = f(x)$ in Case ① and $c_r(x) =$ *True*. Still by this condition, we know $[\forall x' \in \mathbb{A}_{\mathbb{P}}(x), \exists \mathrm{M}_1 \in \mathbb{M}_{\mathbb{P}}, \forall \mathrm{M}_2, \mathrm{M}_3, \mathrm{M}_4 \in \mathbb{M}, f(x' \odot \mathrm{M}_1 \odot \mathrm{M}_2 \odot \mathrm{M}_3 \odot \mathrm{M}_4) = f(x)]$, since $[\exists \mathrm{M}_1 \in \mathbb{M}_{\mathbb{P}}, \forall x' \in \mathbb{A}_{\mathbb{P}}(x), x' \odot \mathrm{M}_1 = x \odot \mathrm{M}_1]$ (see Fig. 2 for illustration). Then by Lemma 2, we know $[\forall x' \in \mathbb{A}_{\mathbb{P}}(x), c_r(x') =$ *True*$]$. By Lemma 1, we also know $[\forall x' \in \mathbb{A}_{\mathbb{P}}(x), g(x') = g(x)]$. Finally, we know $[\forall x' \in \mathbb{A}_{\mathbb{P}}(x), g(x') = g(x) \wedge c_r(x') =$ *True* $\wedge\, c_r(x) =$ *True*$]$. □

B The design subtlety of MRCert

The ordering of the Cases. The sequence of the Cases is crucial. Changing the order may either break the certification guarantee for adversarially patched samples or degrade clean accuracy. For example, swapping Case ② and Case ③ would require the benign counterpart of an adversarially patched sample to exhibit consistency under four masks (rather than the current three) to remain certified; otherwise, an attacker could exploit a fourth-order masked mutant that predicts another label to evade detection. Case 1 is intentionally placed first to prevent benign samples from reaching the recovery stages, thereby improving clean accuracy.

The ordering of mask conditions within a Case. The internal ordering of masking conditions also matters. Consider Case 2 (indirect testing): if its quantifiers were reordered from "there exists mask 1 such that for all mask 2, mask3 ..." to "there exist mask 1 and mask 2 such that for all mask 3 ...", Then adversarial samples containing two patches could escape certification, even when their benign counterparts are round-trip certified. In this reordered structure, the adversarial patched sample could simply choose two masks (mask 1 and mask 2) that expose both patches, i.e., do not cover either patch. Then, even if mask 3 covers one patch, the other patch remains uncovered, allowing the attacker to arbitrarily alter the prediction and evade detection. This demonstrates that the original quantifier ordering is essential for preventing such escape cases.

The number of masks within Cases. Taking Case 2 as an example, the number of masks is also critical. Reducing the number of masks in Case 2 would allow adversarial samples with two patches to escape, for reasons analogous to the ordering issue discussed above. Conversely, increasing the number of masks (e.g., increasing 1 mask) would enlarge the masking requirements for certification. In that case, certifying a benign sample would require prediction consistency under four masks, instead of the current three, which directly decreases the samples that can be certified.

C Extension of MRCert to Recover and Certify Samples with N-patches

We can extend the maximum number of patches from 2 to N (called MRCert-N-patch, a variant of MRCert) following the following idea. First, we apply each set of N masks in the covering mask set $\mathbb{M}$ on the input sample $\hat{x}$ to test whether $\hat{x}$ is harmful. If it is not harmful, MRCert-N-patch returns the label $f(\hat{x})$ (marked as N-Case ①). If $\hat{x}$ is detected as harmful, we then test whether all its first-order mutants are harmful by applying each possible subset with N masks selected with replacement from $\mathbb{M}$ on each first-order mutant of $\hat{x}$. If there exists a first-order mutant, whose all $(N+1)$th-order mutants generated from the first-order mutant of $\hat{x}$ are predicted with the same label as this first-order sample, then $\hat{x}$ is deemed as a one-patch harmful sample, this first-order mutant is "clean" and the prediction label of this first-order mutant is returned (marked as N-Case ②). If that is not the case, we then test whether all second-order mutants of $\hat{x}$ are harmful in the same manner, and repeat until the Nth-order mutants of $\hat{x}$ are tested. For the certification function c_r with the input sample $\hat{x}$, it should be extended to the condition that all $(N+1)$th-order mutants of $\hat{x}$ are predicted with the same label as $\hat{x}$ (for those input samples whose label returned in Case ①), and the condition that there exists a first-order mutant of $\hat{x}$, whose all $(N+2)$th-order mutants are predicted with the same label as this first-order mutant of $\hat{x}$ (for those input samples whose label returned in Case ②), and certifying the input samples output in other cases by the condition in the same manner. For the round-trip certification function, it should be the condition that all $2N$th-order mutants of a benign sample x are predicted with the same label as x. We leave the formal proof and implementation as future work.

D Experimental Setup

D.1 Environment and Dataset

The evaluation is conducted on an Ubuntu 20.04 machine equipped with four Nvidia 3090 GPUs. Following [Xiang *et al.*, 2022], we adopt 1000-class ImageNet [Deng *et al.*, 2009], 10-class CIFAR10 [Krizhevsky *et al.*, 2009], and 10-class ImageNette as our datasets, encompassing both large-scale, diverse datasets and efficient benchmarks widely used for rapid experimentation.

D.2 Models and Baselines

We adopt the Vision Transformer (ViT) as our base model, which achieves the state-of-the-art in many patch robustness certification defenders [Xiang *et al.*, 2024; Li *et al.*, 2022; Salman *et al.*, 2022; Xiang *et al.*, 2022; Huang *et al.*, 2023].

We adopt the state-of-the-art masking-based defender PatchCURE (**PC**) [Xiang *et al.*, 2024] and state-of-the-art smoothing-based defender VOT-CrossCert (**VOT**) from CrossCert [Cro, 2024; Zhou *et al.*, 2024], which shares the common methodology for smoothing-based recovery with ViP [Li *et al.*, 2022] and S-ViT [Salman *et al.*, 2022]. We arm VOT with the methodology for runtime verification proposed by [Yatsura *et al.*, 2023] as our baseline to compare with **MRCert** in our infrastructure.

Specifically, MRCert adopts the end-to-end ViT-SRF model (setting 14x1-k6[1]) proposed and used by PatchCURE [Xiang *et al.*, 2024] from [Xiang, 2024] with the pre-train weights and setting from MAE [He *et al.*, 2022], so that MRCert can use the same strategy in the experiment as PatchCURE [Xiang *et al.*, 2024] to calculate the covering mask set, generate mutants with their predictions inside the ViT-SRF. We use the training scripts with the settings and hyperparameters from the official repository of PatchCURE [Xiang, 2024] for fine-tuning ViT-SRF, which uses the training samples with their first and second-order mutants only. We then apply the same finetuned ViT-SRF to both MRCert and PatchCURE to ensure fairness. We also adopt this official repository [Xiang, 2024] to implement PatchCURE in our infrastructure. We adopt the same pre-trained weights and settings from MAE [He *et al.*, 2022] for VOT. We follow the fine-tuning settings and hyperparameters from the state-of-the-art S-ViT [Salman *et al.*, 2022] for VOT (note that smoothing-based recovery adopts a different notion of mutants, where a mutant is a slice of a sample; therefore, a different fine-tuning on the base model is needed [Li *et al.*, 2022]). We also adopt the column ablation with the ablation size of 19 pixels from [Salman *et al.*, 2022]. We have extracted the

[1]Other variants ViT-SRF14x2, ViT-SRF2x2, and BagNet33 cannot handle the situation against two patches, since all their receptive fields would be inherently masked by VIT-SRF [Xiang, 2024] when generating corresponding mutants. k is the parameter to control the position of splitting SRF and LRF in VIT-SRF in the range [0, 12], which can tune the trade-off between computational efficiency and robustness. We adopt the middle one since it is not the focus of this paper.

corresponding results from the original papers of ViP [Li *et al.*, 2022] and S-ViT [Salman *et al.*, 2022] to compare with VOT in Tab. 1 (their results on round-trip certification are not reported in their papers).

D.3 Metrics

Let x be a benign sample with the true label y_0 in a clean test dataset $\mathbb{D}$, and $R = \langle g(x), c(x), c_r^2(x) \rangle$ be a certified recovery defender. **Clean accuracy** is the fraction of $\mathbb{D}$ that are correctly predicted, defined as $acc_{clean} = \frac{|\{x \in \mathbb{D} | g(x) = y_0\}|}{|\mathbb{D}|}$, which evaluates the standard performance of certified recovery as a classifier. **Certified accuracy** is the fraction of $\mathbb{D}$ that are correctly predicted and certified robust, whose adversarially patched samples should be predicted with the same label as the benign sample, defined as $acc_{cert} = \frac{|\{x \in \mathbb{D} | g(x) = y_0 \wedge c_r(x) = \mathit{True}\}|}{|\mathbb{D}|}$. **Round-trip certified accuracy** is the fraction of $\mathbb{D}$ that are correctly predicted and round-trip certified, whose adversarially patched samples should be predicted with the same label as the benign sample and gain a provable robust verdict, defined as $acc_{cert^2} = \frac{|\{x \in \mathbb{D} | g(x) = y_0 \wedge c_r^2(x) = \mathit{True}\}|}{|\mathbb{D}|}$. The clean accuracy of the base classification model f is $\frac{|\{x \in \mathbb{D} | f(x) = y_0\}|}{|\mathbb{D}|}$. Clean accuracy and certified accuracy are commonly adopted by peers [Levine and Feizi, 2020a; Salman *et al.*, 2022; Li *et al.*, 2022; Xiang *et al.*, 2024; Xiang *et al.*, 2022; Xiang *et al.*, 2021], and round-trip certified accuracy is proposed by [Yatsura *et al.*, 2023].

In RQ1, the test dataset $\mathbb{D}$ is a set of patched samples from attacking a subset of the test dataset, denoted as $\mathbb{D}_{pat}$. Clean accuracy and certified accuracy are measured in this set of adversarially patched samples to evaluate certified recovery defenders on their ability to recover predictions and recover predictions with a provably robust verdict, respectively. In RQ2, we use the original test dataset as $\mathbb{D}$.

D.4 Experimental Procedure

We adopt the patch sizes of 16 pixels (measured as a square patch region with a side length of 16 pixels) and 32 pixels for all three datasets. All samples are rescaled to 224x224 [Zhou *et al.*, 2024; Li *et al.*, 2022; Xiang *et al.*, 2022; Xiang *et al.*, 2024].

In RQ1, we perform an actual adversarial patch attack IFGSM adopted from [Levine and Feizi, 2020a] on PC and MRCert, which uses their shared base model as the attack model for gradient-based attacks without the knowledge of the non-differentiable label recovery function for fairness. Note that the recent proposed attacks are more toward practical, such as limiting the access times or can only get the returned prediction label, while our IFGSM attack has direct and full access to the base model, which is a powerful exam against attackers. We set 80 random starts, 150 iterations per random start, and a step size of 0.05 following [Levine and Feizi, 2020a]. We randomly select 500 test samples from each test dataset for attacks. We follow the practice in [Levine and Feizi, 2020a] to return the worst patched sample for each benign sample and place it into $\mathbb{D}_{pat}$.

In both RQ1 (as a follow-up to the attack) and RQ2, for each defender on each sample in each test dataset, we first generate and evaluate corresponding mutants using the corresponding base model (PC and MRCert share the same ones), then apply each defender to the prediction results collected for these mutants, and measure the metric values.