Different Target Models

Table 4: ViT-b16, 50 runs

	EAC	DeepLIFT	GradSHAP	IntGrad	KernelSHAP	FeatAbl	LIME
ImageNet/Insertion ↑	**89.594**	54.455	68.125	69.480	75.152	65.656	76.161
CoCo/Insertion ↑	**76.759**	37.659	48.888	50.323	63.503	59.072	64.244
ImageNet/Deletion ↓	**17.298**	40.784	30.948	29.903	21.825	34.191	19.254
CoCo/Deletion ↓	**8.318**	28.762	18.422	17.440	9.950	15.946	8.426

Table 5: MobileNet-v2, 50 runs

	EAC	DeepLIFT	GradSHAP	IntGrad	KernelSHAP	FeatAbl	LIME
ImageNet/Insertion ↑	**74.651**	34.197	47.848	48.662	60.837	59.197	61.282
CoCo/Insertion ↑	**68.556**	28.951	37.393	37.719	48.658	44.420	50.387
ImageNet/Deletion ↓	**6.002**	26.381	14.679	13.382	7.766	8.866	7.344
CoCo/Deletion ↓	**6.684**	21.467	14.237	14.936	9.308	11.706	7.106

Table 6: ResNet-18, 50 runs

	EAC	DeepLIFT	GradSHAP	IntGrad	KernelSHAP	FeatAbl	LIME
ImageNet/Insertion ↑	**73.558**	47.799	38.877	36.806	50.547	43.448	50.592
CoCo/Insertion ↑	**65.669**	50.689	42.937	45.252	54.046	53.835	53.837
ImageNet/Deletion ↓	**6.596**	8.588	11.273	11.555	6.638	8.352	6.776
CoCo/Deletion ↓	**5.015**	9.097	11.758	11.483	7.007	9.325	6.495

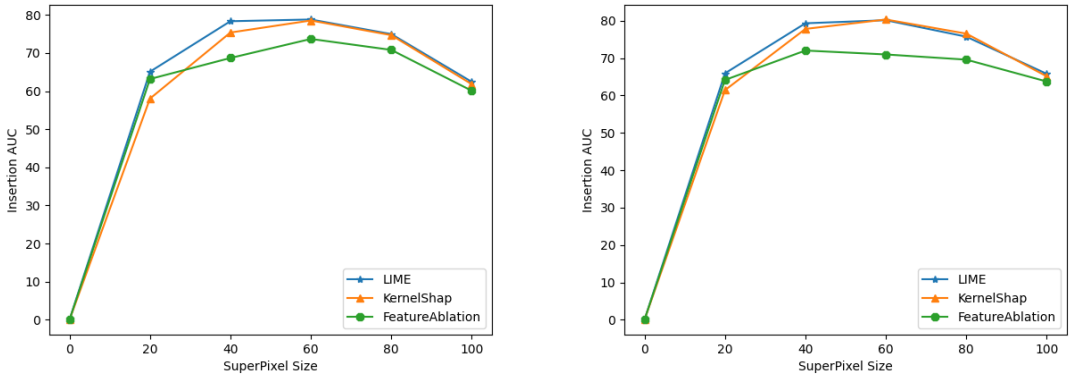
We explore the performance of EAC on different target models. We choose three representative visual models, including ViT [48], MobileNet [49], and ResNet-18 [1], and use the same experimental setup as in the main text. We run each method for 50 times to report the average performance of each method. Overall, we observe a similar performance as shown in the main text. In particular, EAC consistently outperforms other methods on all target models.

Backdoor Defense

Table 7: Backdoor-Defense on CIFAR-10

ASR	Victim Model	EAC	DeepLIFT	GradSHAP	IntGrad	KernelSHAP	FeatAbl	LIME
BadNet [50] ↓	0.99	**0.042**	0.542	0.622	0.618	0.91	0.47	0.574
TrojanNN [51] ↓	0.99	**0.038**	0.094	0.122	0.122	0.65	0.098	0.11

To evaluate the security impact of EAC, this section conducts backdoor removal experiments on CIFAR-10 [52]. We compare EAC and other XAI methods. Specifically, we perform two representative backdoor attacks, BadNet [50] and TrojanNN [51], on ResNet-18 as the target model. During the evaluation process, aligned with relevant works in this field [53], we remove the top three patches among every poisoned image for each XAI tool, and record the corresponding Attack Success Rate (ASR) after the removal. Overall, we randomly generate 250 poisoned images, and report their average ASR in Table 7. The evaluation results are highly encouraging; EAC has the lowest ASR under both attack settings. We interpret that this evaluation shows the high generalizability of EAC over different target models.



(a) COCO, 100 runs

(b) ImageNet, 100 runs

Figure 3: The effect of the superpixel size on AUC. A trade-off can be observed.

Analysis of the Trade-off between SuperPixel Size and AUC

Overall, superpixel-based XAI tools are sensitive to the size of the superpixel. To obtain a fair comparison between EAC and de facto superpixel-based XAI tools, we carefully studied how the size of superpixel influence the performance of LIME, KernelShap, and FeatureAblation. The evaluation results using both ImageNet and COCO are shown in Fig. 3. We observed that there exists a trade-off between AUC and the superpixel size for both datasets. Empirical observation shows that a proper range of the superpixel size ranges from 40 to 80.

To unleash the full capability of superpixel-based methods, we set the superpixel size as 75 for ImageNet evaluations, and 58 for COCO evaluations, respectively, when conducting the experiments in the main paper. In contrast, EAC does *not* require such a hyperparameter tuning step, and is able to achieve superior performance over those superpixel-based methods.