REAP: RARE-EVENT ADAPTATION FOR DEEP LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Rare but important samples are critical to model reliability, yet their low frequency creates a persistent rare-sample underfitting problem: standard training is dominated by common patterns, causing rare samples to be systematically underrepresented and poorly learned. Existing importance sampling strategies can increase the exposure of informative samples during post-training, but they do not modify the model architecture to help pretrained models adapt to rare samples, motivating an architectural solution for more effective rare-sample adaptation. To address this challenge, we propose REAP, a rarity-aware adapter that can be embedded into pretrained backbone networks during post-training to facilitate adaptation to rare samples. REAP is applicable to multi-stage hierarchical backbones and introduces lightweight residual adaptation, enabling a smooth transition from pretrained representations to post-training. Furthermore, we propose an implicit routing mechanism, where continuous spatial gates modulate adapter residuals through element-wise multiplication. This design enables smooth and stable adaptation without requiring additional supervision signals or rule-based preprocessing, thereby supporting fully end-to-end post-training. REAP can also be naturally combined with existing importance sampling methods, yielding complementary benefits from architectural adaptation and informative-sample sampling. Extensive experiments with a SegFormer backbone across semantic segmentation, medical image classification, and regression tasks demonstrate the effectiveness of REAP. Compared with the corresponding no-Adapter baselines, REAP improves tail IoU on ADE20K by 10.64%, increases classification accuracy on ISIC by 3.64%, and reduces the PHM08 score on the FD001 dataset by 7.12%, while preserving overall performance, contributing to safer and more trustworthy AI systems.

1 INTRODUCTION

Rare but informative samples are often crucial for the reliability of learning systems, yet their low frequency gives rise to a broader issue of **rare-sample underfitting**: because standard training is dominated by common patterns, rare samples contribute weak and noisy learning signals, causing the model to systematically underfit them (Menon et al., 2021). Although this was first highlighted in safety-critical domains like autonomous driving and medical diagnosis, it is equally relevant to general machine learning tasks whenever rare samples carry disproportionate importance but receive insufficient gradient signal due to their low sampling probability (Katharopoulos & Fleuret, 2018). In semantic segmentation, for instance, infrequently occurring object categories are poorly predicted; in medical imaging, rare but critical pathological patterns are overshadowed by normal cases; and in general regression tasks, samples from extreme regimes are often underfit by the model, despite their practical significance. We refer to this systematic underfitting of rare but important samples as the **curse of rarity** (Liu & Feng, 2024). It represents a fundamental obstacle to building trustworthy AI systems that perform reliably across the full spectrum of data distributions.

A prevalent line of research addresses this challenge via importance sampling (Zhao & Zhang, 2015), which reweights or resamples the training distribution to amplify the exposure of informative samples during post-training. Gradient-based importance sampling assigns higher sampling probabilities to samples with larger gradient norms, thereby allocating more optimization effort to them and reducing the variance of gradient estimates (Johnson & Guestrin, 2018). These strategies have

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.

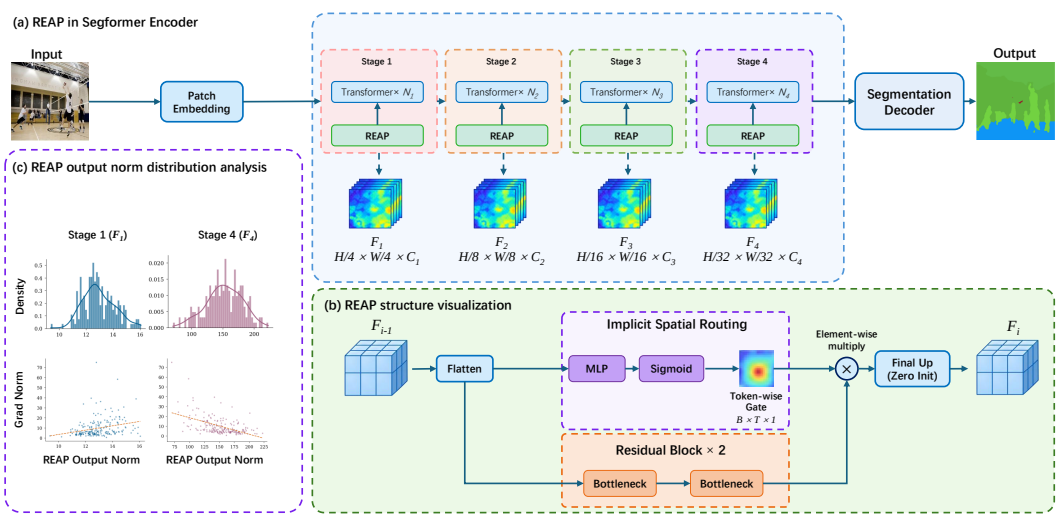

Figure 1: **Overview of REAP framework. (a):** Integration of REAP modules into the hierarchical SegFormer encoder across four feature stages (F_1 to F_4). **(b):** Detailed architecture of REAP. **(c):** REAP output norm distribution analysis. Norm distribution reveals that most samples receive small corrections while rare samples are heavily modified; Across layers, shallow adapters (F_1) actively correct for high-gradient samples, whereas deep adapters (F_4) adopt a conservative strategy.

yielded empirical gains in long-tail learning tasks, highlighting the importance of controlling sample exposure during training. In particular, adaptive sampling can allocate more training updates to informative but underrepresented examples, which is especially relevant to rare-event learning. While importance sampling improves gradient efficiency by upweighting high-gradient samples, it remains a purely data-centric intervention that leaves the model's representational capacity unchanged. Because pretrained models remain biased toward dominant modes, increased exposure alone cannot overcome the absence of dedicated pathways for processing low-frequency samples (Pan et al., 2025). Overcoming this constraint therefore requires **coupling data-centric sampling with architectural adaptations**: importance sampling decides which samples receive optimization focus, whereas a rarity-aware architecture determines how the network processes them.

To bridge this gap, we propose **REAP**, a parameter-efficient adapter designed for hierarchical architectures that mitigates the curse of rarity in post-training. By injecting lightweight residual pathways across feature stages with identity-initialized projections, REAP enables specialized adaptation to rare samples while preserving pretrained representations and ensuring stable optimization.

A central design in REAP is its implicit routing mechanism. Rather than relying on hard routing decisions or external annotations, REAP predicts a continuous spatial gate from the input feature and uses it to modulate the adapter residual through element-wise multiplication. Let $z_l \in \mathbb{R}^{B\times C_l\times H_l\times W_l}$ denote the feature map at stage l. After flattening the spatial dimensions into tokens, the gate is computed as $g_l = \sigma(f_l(z_l)) \in (0,1)^{B\times H_l W_l\times 1}$, and the adapted feature is given by $\tilde{z}_l = z_l + g_l \odot U_l(h_l(z_l; g_l))$, where h_l denotes the stack of gated bottleneck residual blocks and U_l is the zero-initialized final projection. This design has two advantages: first, it requires no external supervision or rule-based preprocessing, as the gate is learned end-to-end from the task loss; Second, its continuous and differentiable formulation provides smooth gradient propagation. In this way, REAP learns to identify spatially informative or atypical regions and adaptively modulates the corresponding residual corrections within a single end-to-end optimization procedure.

REAP further integrates naturally with existing gradient-based importance sampling, yielding benefits from two orthogonal perspectives: the sampler increases the exposure of informative samples at the data level, while the adapter provides the architectural capacity at the structural level. Neither mechanism alone fully resolves the curse of rarity: sampling without architectural adaptation leaves the model structurally biased, while adaptation without targeted sampling provides insufficient training signal. Their combination, however, creates a synergistic loop in which rare samples are both sampled more frequently and learned more effectively.

Extensive experiments on general vision tasks such as ImageNet (Deng et al., 2009) classification and ADE20K (Zhou et al., 2017) scene parsing, medical imaging tasks like Kvasir-SEG (Jha et al., 2020) polyp segmentation and ISIC 2018 (Codella et al., 2019) skin lesion classification, and regression tasks like NASA C-MAPSS (Saxena et al., 2008a) remaining useful life (RUL) prediction demonstrate that REAP improves rare-sample performance while preserving overall performance gained from pre-training. Our contributions are summarized as follows:

- We propose REAP, a parameter-efficient adapter module integrated into pretrained multi-stage backbones for rare-sample adaptation and enabling a smooth transition from the pre-trained representation during post-training.
- We introduce an implicit routing mechanism based on continuous spatial gating and element-wise modulation. Learned end-to-end, this mechanism adaptively modulates residual corrections at different spatial locations without requiring additional supervision.
- We show that REAP can be naturally combined with existing gradient-based importance sampling, yielding complementary benefits: importance sampling increases exposure to informative samples, while REAP provides dedicated capacity for adapting to them.
- We conduct extensive experiments across general vision, medical imaging, and regression tasks, demonstrating the generality of the proposed approach and its potential contribution to more trustworthy learning systems.

2 Related Works

2.1 Curse of Rarity and Rare-Sample Learning

Rare but safety-critical samples present a fundamental challenge for robust learning systems. This phenomenon, known as the **curse of rarity**, was first formalized in the context of autonomous driving, where safety-critical scenarios occur infrequently but have disproportionate consequences (Liu & Feng, 2024); standard empirical-risk-minimization pipelines therefore tend to underrepresent tail events and generalize poorly to them. Similar long-tailed distributions are prevalent in medical image analysis, industrial defect inspection, and other safety-critical applications, where rare pathological or anomalous cases are often more important than common samples. Because rare samples appear only sparsely during training, their learning signals are easily overwhelmed, resulting in biased decision boundaries and inferior tail performance (Johnson & Khoshgoftaar, 2019). Existing approaches primarily address this issue through data-level or loss-level rebalancing, including synthetic minority oversampling, focal loss, and distribution-aware margins (Zhang et al., 2023). While proven effective on certain areas, these methods mainly modify the sampling distribution or optimization objective and do not fundamentally change the model's structural capacity to learn the rare samples. In contrast, our work investigates the problem from an architectural perspective.

2.2 Importance Sampling and Informative-Sample Selection

Importance sampling and hard-example mining adjust the training distribution by selecting samples with high gradients or great predictive difficulty, such as gradient-based importance sampling (Katharopoulos & Fleuret, 2018) and OHEM (Shrivastava et al., 2016). For learning tasks with long-tailed distributions, class-balanced objectives (Cui et al., 2019), dynamic reweighting (Ren et al., 2018), and distributionally robust optimization —(Samuel & Chechik, 2021) have been used to increase the influence of tail or difficult samples. These methods primarily alter sample exposure or optimization weights rather than the model's internal representation capacity for rare samples. We therefore use gradient-based importance sampling as a representative baseline and introduce REAP to provide dedicated architectural capacity for adapting pretrained models to rare samples.

2.3 Parameter-Efficient Adaptation and Dynamic Routing

Mainstream parameter-efficient transfer learning (PEFT) methods (e.g., LoRA) efficiently adapt frozen backbones but apply static updates across all inputs, ignoring individual sample difficulty (Hu et al., 2022). While conditional computation introduces dynamic routing to enable sample-level adaptive tuning, existing methods typically rely on discrete expert selection (Shazeer et al., 2017) or

explicit routing supervision (Aljundi et al., 2017). In contrast, REAP utilizes a continuous, input-dependent spatial gate to dynamically modulate adapter residuals based on sample difficulty, enabling end-to-end, instance-aware adaptation with minimal parameter overhead.

3 METHODS

The central idea of our approach is to mitigate the curse of rarity from an architectural perspective. During standard training, gradients from common patterns dominate the optimization process, causing the learning signals associated with rare samples to be easily overwhelmed. Even when importance sampling increases the exposure of rare samples, they primarily modify which samples are used for optimization, rather than how the model adapts to different samples. Consequently, increasing the frequency of rare samples alone does not provide a mechanism for allocating adaptation capacity according to each input. This observation suggests that rare-sample learning should be addressed not only through data selection, but also through architectural adaptation. Motivated by this observation, **REAP** is proposed as a parameter-efficient adapter module with an implicit routing mechanism.

REAP adds residual adapters to multi-stage hierarchical backbones and modulates their outputs with a differentiable spatial gate. This enables end-to-end post-training adaptation without additional supervision or rule-based preprocessing. Figure 1 provides an overview of the framework.

3.1 PROBLEM SETUP

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the training set, where x_i is an input sample and y_i is its corresponding supervision signal. We build on a pretrained multi-stage hierarchical backbone E, which produces a set of intermediate feature maps $z_l = E_l(x), l = 1, \ldots, L$,where $z_l \in \mathbb{R}^{B \times C_l \times H_l \times W_l}$ denotes the feature representation at stage l. A decoder D_ϕ maps the backbone features to the final output.

REAP is inserted into the backbone at multiple feature stages. For each stage l, the adapted feature is computed as $\tilde{z}_l = z_l + A_{\psi_l}(z_l)$, where A_{ψ_l} denotes the adapter module at stage l, parameterized by ψ_l. The adapted features $\{\tilde{z}_l\}_{l=1}^{L}$ are then passed to the decoder D_ϕ. The overall predictor can therefore be written as $f_\theta(x) = D_\phi\left(\{\tilde{z}_l\}_{l=1}^{L}\right)$, where $\theta = \{\phi, \psi_1, \ldots, \psi_L\}$ denotes the trainable parameters. The standard post-training objective is $\mathcal{L}_{\text{task}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell\left(f_\theta(x_i), y_i\right)$, where $\ell(\cdot, \cdot)$ is the task-specific loss function.

REAP is designed not to replace standard post-training, but to provide an architectural mechanism that helps pretrained models adapt to rare and challenging samples while preserving their performance on common samples.

3.2 EXISTING GRADIENT-BASED IMPORTANCE SAMPLING

We adopt an existing gradient-based importance sampling strategy during post-training. For each training sample (x_i, y_i), we compute the norm of its per-sample gradient with respect to the trainable parameters: $s_i = \left\|\nabla_\theta \ell\left(f_\theta(x_i), y_i\right)\right\|_2$. Samples with larger gradient norms are typically harder or less fitted by the current model. The sampling probability of each sample is then defined as

$$q_i = \frac{s_i}{\sum_{j=1}^{N} s_j}. \tag{1}$$

During post-training, mini-batches can be sampled according to q_i, or according to a mixture of q_i and the uniform distribution, in order to balance informative-sample exposure and training stability.

We combine the existing importance-sampling (IS) strategy with REAP to make more effective use of the learning signals.

3.3 PARAMETER-EFFICIENT RESIDUAL ADAPTATION

REAP introduces lightweight residual adapters at multiple feature stages of a hierarchical backbone. The following describes how it operates during post-training adaptation.

For a given stage l, let $z_l \in \mathbb{R}^{B\times C_l\times H_l\times W_l}$ be the input feature map. We first reshape it into a token-wise representation $Z_l \in \mathbb{R}^{B\times T_l\times C_l}, T_l = H_l W_l$, where each token corresponds to one spatial location. The adapter branch contains J lightweight bottleneck residual blocks. Each bottleneck block projects the feature to a low-dimensional space and projects it back after activation. The j-th bottleneck transformation can be written as $F_{l,j}(H) = W_{l,j}^{\mathrm{up}}\,\gamma\Big(W_{l,j}^{\mathrm{down}}H\Big)$, where $\gamma(\cdot)$ denotes the activation function, $W_{l,j}^{\mathrm{down}} \in \mathbb{R}^{r_l\times C_l}$ is the down-projection matrix, $W_{l,j}^{\mathrm{up}} \in \mathbb{R}^{C_l\times r_l}$ is the up-projection matrix, and $r_l \ll C_l$ is the bottleneck dimension. This bottleneck design keeps the adapter parameter overhead small relative to full fine-tuning.

Let $H_l^{(0)} = Z_l$ denote the input to the adapter branch. Without routing, the residual transformation at stage l would be computed as $H_l^{(j)} = H_l^{(j-1)} + F_{l,j}\Big(H_l^{(j-1)}\Big), j = 1,\dots,J$. The final adapter residual is then produced by an output projection U_l: $A_l = U_l\Big(H_l^{(J)}\Big)$. To ensure a smooth transition from pretraining to post-training, we initialize the final projection U_l to zero. At the beginning of post-training, $A_l = 0, \tilde{z}_l = z_l$, so the network output is identical to that of the pretrained backbone.

3.4 Implicit Routing via Element-wise Spatial Modulation

A standard adapter shares the same residual transformation across samples. However, rare samples may contain localized or atypical patterns that benefit from spatially selective adaptation. To address this, REAP introduces an implicit routing mechanism based on token-level spatial gating. Unlike MoE selection, our method continuously modulates a shared adapter via locally predicted spatial gates for differentiable, input-dependent adaptation.

For each stage l, we compute a continuous spatial gate from input Z_l: $G_l = \sigma(\psi_l(Z_l))$, where $\psi_l(\cdot)$ is a lightweight multilayer perceptron and $\sigma(\cdot)$ denotes the sigmoid function. The gate G_l has shape $G_l \in [0,1]^{B\times T_l\times 1}$, and is broadcast across the channel dimension. Thus, G_l acts as a token-level spatial gate, controlling how strongly the adapter residual is applied at each spatial location.

The gate is integrated into the adapter residual computation through element-wise multiplication. Specifically, the gated adapter transformation is defined as

$$H_l^{(0)} = Z_l, \tag{2}$$

$$H_l^{(j)} = \Big[H_l^{(j-1)} + F_{l,j}\Big(H_l^{(j-1)}\Big)\Big] \odot G_l, \qquad j = 1,\dots,J, \tag{3}$$

$$A_l = U_l\Big(H_l^{(J)}\Big) \odot G_l, \tag{4}$$

where $\odot$ denotes element-wise multiplication. The adapted feature is then obtained by adding the gated adapter residual back to the original backbone feature: $\tilde{Z}_l = Z_l + A_l$. After reshaping $\tilde{Z}_l$ back to a feature map, we obtain $\tilde{z}_l$.

This design requires no auxiliary routing supervision or external rarity labels, and uses continuous spatial modulation with a zero-initialized projection to preserve the pretrained representation. The router does not explicitly predict sample-level rarity; instead, when combined with IS, it encounters rare samples more frequently and learns their signals through the task loss. As observed in Figure 1c, shallow adapters (F_1) apply stronger corrections to high-gradient samples, as fine-grained features support localized adaptation, whereas deep adapters (F_4) remain conservative because compressed semantic representations may make aggressive updates more globally disruptive.

3.5 Training Protocol

During post-training, the encoder, decoder, and adapter parameters are optimized jointly using the task loss:

$$\mathcal{L}_{\mathcal{B}}(\theta) = \frac{1}{|\mathcal{B}|}\sum_{(x_i,y_i)\in\mathcal{B}} \ell\Big(D_\phi\Big(\{z_l + A_{\psi_l}(z_l)\}_{l=1}^{L}\Big), y_i\Big), \tag{5}$$

where z_l denotes the feature at stage l, and θ includes the encoder, decoder, and adapter parameters. Mini-batches are sampled using the IS strategy in Equation 1, optionally mixed with random samples.

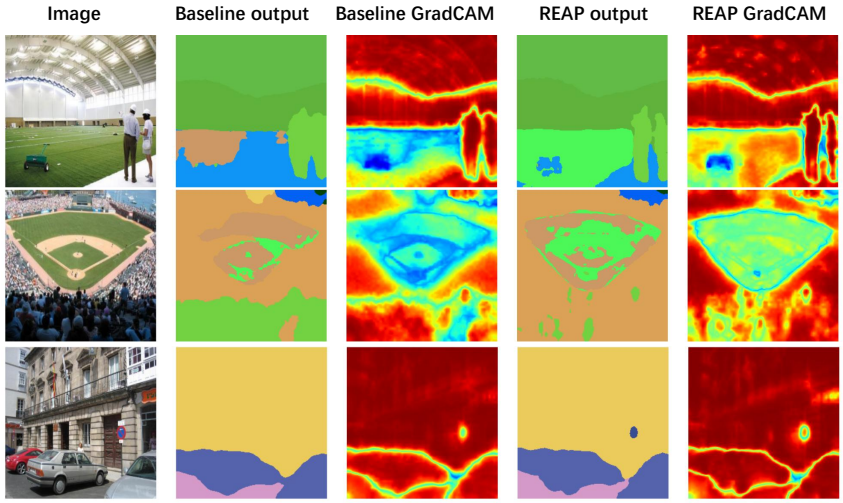

Figure 2: **Qualitative visualization on the ADE20K dataset.** Visual comparisons demonstrate three representative scenarios: person, stadium, and car. Compared to the baseline, REAP shows significant improvements, including accurately producing finer boundaries of humans, successfully understanding complex stadium structures, and precisely capturing vehicle contours while detecting street signboards missed by the baseline.

4 Experiments

4.1 Experimental Setup

Tasks and datasets. We evaluate REAP on classification, segmentation, and regression to assess its generality across various tasks. We evaluate classification on ISIC 2018 (skin lesion classification), Kvasir (polyp classification) (Pogorelov et al., 2017), and a long-tailed ImageNet subset; segmentation on Kvasir-SEG (polyp segmentation) and ADE20K (semantic scene parsing); and regression on C-MAPSS FD001/FD004 (turbofan engine RUL prediction) and freMTPL2 (motor insurance claim regression) (Noll et al., 2020). Dataset statistics are provided in Appendix A.1.

Model and training. We use SegFormer (Xie et al., 2021) as the hierarchical encoder and insert REAP modules at every encoder stage. All components are optimized end-to-end with task-specific objectives.

Baselines and evaluation. We compare REAP with importance sampling (REAP+IS) against standard post-training and importance sampling. We report both overall task performance (e.g., F1-score) and rare-sample performance (e.g., Dice on Rare Samples). Implementation details are given in Appendix A.2, visualization in Figure 2 and Appendix A.3.

4.2 Comparing with Existing Methods

We compare REAP+IS with representative state-of-the-art methods on the ISIC 2018 and Kvasir datasets. We report F1-score and overall accuracy (OA) as evaluation metrics. As shown in Table 1, our method achieves state-of-the-art performance on both benchmarks.

4.3 Ablation Study Across Different Tasks

To quantify the individual and combined contributions of our proposed components, we conduct an ablation study across all three task families. Specifically, we compare three models: SegFormer

(Base), IS, and IS+REAP (Ours). The comparison between SegFormer (Base) and IS isolates the effect of importance sampling, while the comparison between IS and IS+REAP measures the structural contribution of REAP. This progressive evaluation demonstrates the respective and combined benefits of importance sampling and REAP, highlighting the complementarity between data-level sample selection and architectural adaptation.

Table 1: Performance comparison on ISIC and Kvasir. Best and second-best results are shown in **bold** and underlined. OA: overall accuracy.

ISIC 2018			Kvasir		
Model	F1 (%) ↑	OA (%) ↑	Model	F1 (%) ↑	OA (%) ↑
Swin-T	61.9	66.1	Swin-S	77.3	78.0
ConvNeXt-T	63.2	67.1	ConvNeXt-S	74.8	74.8
Repvgg-A1	68.3	71.6	TNT-S	75.7	76.2
MobilevitV2	65.2	68.1	Caformer-S	73.5	73.7
EdgeNeXt-B	64.5	67.7	Deit-S	77.9	78.1
Nest-T	64.2	69.1	EfficientNetV2-S	78.1	78.2
MobileOne-S4	70.0	72.2	Coat-S	73.1	73.5
CaiT	58.4	63.9	VMamba-S	77.1	77.3
VMamba-T	70.3	72.5	HiFuse-S	80.5	81.0
HiFuse-T	73.9	75.6	MedMamba-S	79.2	79.3
MedMamba-T	72.3	74.0	MedViT-V1-S	79.6	80.2
MedViT-V1-T	69.4	72.4	MedViT-V2-S	82.5	82.8
MedViT-V2-T	76.2	77.1	SegFormer	93.75	**93.75**
SegFormer	69.3	84.4			
REAP+IS	**78.3**	**88.0**	**REAP+IS**	**93.76**	**93.75**

Table 2: Performance comparison on ISIC and Kvasir. Acc: Accuracy; F1: F1-Score; AUC: Area Under the ROC Curve; P@0.99R: Precision at 99% Recall.

		Acc	F1	AUC	P@ 0.99R
ISIC	Base	0.8440	0.6931	0.9641	0.2043
	IS	0.8544	0.7170	**0.9667**	**0.2070**
	REAP	**0.8804**	**0.7833**	0.9629	0.1999
		Acc	F1	AUC	P@ 0.99R
Kvasir	Base	**0.9375**	0.9375	0.9954	0.7687
	IS	0.9325	0.9325	0.9952	0.7939
	REAP	**0.9375**	**0.9376**	**0.9960**	**0.7975**

4.3.1 IMAGE CLASSIFICATION

We conduct the classification ablation on ISIC 2018, Kvasir, and a self-constructed long-tailed ImageNet subset, comparing SegFormer (Base), importance sampling (IS), and IS with REAP (REAP). We select the checkpoint with the best $\mathrm{P@0.99R}$ on ISIC 2018 and Kvasir. For the ImageNet subset, we augment one class using AutoAugment while leaving the other two unchanged, and then select samples to construct a severely imbalanced subset with a class ratio of 100:1:1. Given this extreme imbalance, $\mathrm{P@0.99R}$ is highly sensitive to small changes in minority-class predictions; therefore, we use the best $\mathrm{P@0.9R}$ for more robust checkpoint selection.

As shown in Table 2 and Table 4, REAP achieves consistent improvements over the Base model. On Kvasir, REAP obtains the best F1-score, AUC, and $\mathrm{P@0.99R}$, while maintaining accuracy. On ISIC 2018, REAP improves accuracy and F1-score. On the self-constructed long-tailed ImageNet subset, REAP achieves the best $\mathrm{P@0.99R}$ and F1-score. These results indicate that importance sampling and REAP provide complementary benefits: IS improves the exposure to rare samples, whereas REAP enhances the model's structural ability to adapt to them.

4.3.2 SEMANTIC SEGMENTATION AND SCENE PARSING

We evaluate semantic segmentation performance on ADE20K, a benchmark for general multi-class scene parsing, and on Kvasir-SEG, a dataset for polyp segmentation. We use $\mathrm{P@0.99R}$, IoU, and Dice as the primary metrics, and additionally report tail-oriented metrics that reflect performance on rare samples.

As shown in Table 3 and Table 5, IS alone yields suboptimal performance on dense prediction tasks, even underperforming the Base model on ADE20K. This degradation stems from two main factors. First, computing the L_2 norm of per-sample gradients across entire images causes sampling probabilities to be dominated by complex backgrounds, forcing the model to overfit unlearnable patterns. Second, common and rare classes co-occur within the same image; IS suppresses the optimization of common categories, leading to drops in Head and Medium IoU.

In contrast, integrating REAP with IS consistently outperforms both the Base model and IS. By introducing lightweight adapters with spatial gating, REAP resolves the limitations of data-centric

Table 3: Performance comparison on ADE20K. Head, medium, and tail IoU denote the mean per-class IoU for classes grouped by pixel frequency r: head ($r \geq 1\%$), medium ($0.1\% \leq r < 1\%$), and tail ($r < 0.1\%$). HTGap is the head-tail IoU difference. Best results are in **bold**; second best underlined.

Method	mIoU	Dice	PixelAcc	P@0.99R	P@0.95R	P@0.9R	Head IoU	Med IoU	Tail IoU	HTGap↓
Base	0.2414	0.3360	0.7505	0.0680	0.1254	0.1701	0.5387	0.3229	0.0994	0.4392
IS	0.2083	0.2879	0.7390	0.0584	0.1133	0.1549	0.5256	0.2809	0.0675	**0.4581**
REAP	**0.3061**	**0.4316**	**0.7575**	**0.0716**	**0.1372**	**0.1868**	**0.5446**	**0.3533**	**0.2058**	0.3388

sampling from an architectural perspective by adaptively allocating spatial capacity to modulate feature representations. This absorbs the high-gradient signals provided by IS, achieving a substantial +13.83% increase in Tail IoU on ADE20K and marked gains in Rare Dice on Kvasir-SEG while maintaining performance on common categories. Figure 2 illustrates qualitative visual comparisons and GradCAM heatmaps between REAP and baseline on ADE20K.

4.3.3 Regression across Industrial and Insurance Domains

To evaluate the generalizability of REAP beyond visual tasks, we conduct experiments in two regression domains: remaining useful life (RUL) prediction for industrial turbofan engines on C-MAPSS, and motor insurance claim modeling on freMTPL2. In both settings, empirical risk minimization (ERM) tends to underfit rare regimes, despite their disproportionate importance.

To address this, we employ a two-stage post-training paradigm centered around a *Top-Gradient Key Sampling* protocol. After initializing from a checkpoint, we compute the per-sample gradient ℓ_2-norm with respect to the trainable parameters: $s_i = \|\nabla_\theta \mathcal{L}(f_\theta(x_i), y_i)\|_2$.

Table 4: Performance comparison on ImageNet subset. **P@0.99R**: Precision at 99% Recall; **P@0.9R**: Precision at 90% Recall; **F1**: F1-Score. Best results are in **bold**; second best underlined.

Model	P@0.99R	P@0.9R	F1
Base	0.0805	0.5838	0.8230
IS	0.2610	**0.7056**	0.8836
REAP	**0.3234**	0.6487	**0.8874**

Table 5: Segmentation results on Kvasir-SEG. Rare Dice: mean per-sample Dice over the top 25% samples ranked by per-sample gradient norm; P@0.99R: precision at 99% recall. Best results are in **bold**; second best underlined.

Method	mIoU	Dice	P@0.99R	Rare Dice
Base	0.8339	0.8946	0.1862	0.7445
IS	0.8325	0.8933	0.1939	0.8305
REAP	**0.8399**	**0.9014**	**0.2846**	**0.8459**

We then select the top-K samples with the highest gradient norms ($K = \alpha N$, where α denotes the sampling fraction) to form the "key set," and perform fine-tuning exclusively on these highly informative instances. To account for the evolving model capacity, we introduce a *dynamic rescoring* mechanism that periodically refreshes the key set during post-training. The domain-specific configurations are detailed as follows:

C-MAPSS (Industrial RUL Prediction). For turbofan engine degradation modeling, we fix the sampling fraction to $\alpha = 0.01$, thereby selecting the top 1% most critical samples. Gradients are computed using the per-sample squared error loss. The key set is processed in a single forward-backward pass per optimization step, with dynamic rescoring optionally applied. To account for varying data scales, we fine-tune the model for 20 steps on FD001 and for 200 on FD004, which exhibits a substantially larger volume of trajectory records. The optimization objective is the root mean squared error (RMSE) evaluated over the key set.

freMTPL2 (Insurance Claim Modeling). For the French motor third-party liability dataset, the task requires joint estimation of claim severity. Owing to the larger dataset scale and high variance in the loss landscape, we set $\alpha = 0.02$ and train for 100 steps. Dynamic rescoring is executed at every step to rapidly adapt to the heavy-tailed distribution. The training objective is weighted log-squared error for claim severity. Due to memory constraints, gradients are accumulated via micro-batching.

Evaluation Metrics. To rigorously assess overall accuracy alongside fidelity in safety- and financially critical regimes, we adopt domain-specific metrics:

- **C-MAPSS Metrics:** Predictions and ground-truth values are first mapped back to the raw operational cycles ($y^{\text{raw}} = 125 \times y$, with predictions non-negatively clamped). We report: (i) *Overall RMSE*, measuring global RUL estimation error; (ii) *Low-RUL RMSE*, which evaluates prediction error strictly on near-failure units ($y^{\text{raw}} < 20$ cycles), representing the most critical operational regime in predictive maintenance; (iii) *PHM08 Score (Saxena et al., 2008b)*.

Table 6: Comparison of Base and REAP methods on FD001 and FD004 datasets.

		rmse	PHM08	Low Rul rmse
FD001	Base	15.1328	404.6277	**4.3759**
	REAP	**15.0316**	**397.5031**	4.3885
FD004	Base	24.8334	14716.4773	**15.2648**
	REAP	**24.6849**	**13198.4865**	16.9133

Table 7: Regression results on freMTPL2 dataset. All metrics are lower-is-better (↓).

Method	$\text{MSE}_{\log 1p}$ ↓	RMSE↓	MSE↓	MAE↓
Base	1.5652	7109.8	50549268	1675.4938
REAP	**1.4834**	**7109.62**	**50546664**	**1588.5901**

- **freMTPL2 Metrics:** We evaluate the severity branch on policies with positive claims ($N_i > 0$). Ground-truth severity is defined as $y_i = A_i/N_i$, weighted by $w_i = N_i$, where N_i and A_i denote the claim count and total claim amount. We report:
(i) *Weighted RMSE, MSE, and MAE*; (ii) *Weighted MSE in* $\log 1p$ *space*. Details of these metrics are shown in A.2.2.

As shown in Table 6 and Table 7, REAP consistently improves the pretrained baseline across both regression tasks. On C-MAPSS, REAP achieves lower overall RMSE and reductions in the PHM08 score on both subsets—most notably on the more complex FD004 regime, where the score drops from 14716.48 to 13198.49, demonstrating that the model effectively penalizes safety-critical late predictions despite a slight trade-off in Low-RUL RMSE. On freMTPL2, REAP reduces the key evaluation metric, $\text{MSE}_{\log 1p}$, as well as MAE, while remaining competitive on raw MSE and RMSE. These results confirm that REAP generalizes effectively beyond visual domains to regression problems dominated by high-impact and rare events.

5 Conclusion

We presented REAP, a parameter-efficient adapter that mitigates the curse of rarity from an architectural perspective. By introducing lightweight adapters with continuous spatial gating, REAP enables sample-conditioned adaptation without rule-based preprocessing. The initialization ensures a smooth transition from pretraining, while implicit routing allows the model to modulate adapter residuals at informative spatial locations through end-to-end optimization. Furthermore, REAP integrates naturally with existing importance sampling, combining data-level optimization with structural adaptation capacity. Extensive experiments demonstrate consistent improvements on performance metrics, suggesting that addressing rare-sample underfitting requires both data selection and architectural mechanisms, and that their synergy contributes to more reliable and trustworthy learning systems.

AI use statement

In this work, we used generative AI tools for refining language expression and assisting with experimental code debugging. We have not used generative AI tools for research idea generation, core algorithm design, or writing original manuscript content, and other generative tasks are not applicable to this work.

Additionally, we reviewed all AI-assisted work thoroughly. Specifically, language refinements were carefully verified by the authors to ensure scientific accuracy and proper terminology, and AI-assisted code for debugging was manually tested and validated through experimental runs. We take full responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics statement

This paper presents a general structural framework and does not introduce specific ethical risks or social harms. All benchmarks utilized in our experiments (including ISIC 2018, Kvasir, and Kvasir-SEG) are fully anonymized and publicly available for open academic research. No direct human subjects, proprietary clinical data, or non-public personal information were involved, and no custom IRB authorization was required for this study.

Reproducibility statement

To ensure the reproducibility of our work, we provide complete details of our experimental configurations in the appendix. Specifically, detailed dataset descriptions and statistics are provided in Appendix A.1; hyperparameter choices and implementation details are summarized in Appendix A.2; definitions of evaluation metrics are specified in Appendix A.2.2; and baseline selections are discussed in Appendix A.2.3.

References

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7120–7129, 2017.

Noel Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019. arXiv:1902.03368v2.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. RepVGG: Making VGG-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13728–13737, 2021.

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15908–15919, 2021.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Xiangzuo Huo, Gang Sun, Shengwei Tian, Yan Wang, Long Yu, Jun Long, Wendong Zhang, and Aolun Li. HiFuse: Hierarchical multi-scale feature fusion network for medical image classification. *Biomedical Signal Processing and Control*, 87:105534, 2024.

Debesh Jha, Pia H. Smedsrud, Michael A. Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D. Johansen. Kvasir-seg: A segmented polyp dataset. In *MultiMedia Modeling: 26th International Conference, MMM 2020, Daejeon, South Korea, January 5–8, 2020, Proceedings, Part II*, pp. 451–462, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-37733-5.

Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, March 2019. ISSN 2196-1115.

Tyler B. Johnson and Carlos Guestrin. Training deep models faster with robust, approximate importance sampling. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 7276–7286, 2018.

Angelos Katharopoulos and Francois Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2525–2534, 2018.

Henry X. Liu and Shuo Feng. Curse of rarity for autonomous vehicles. *Nature Communications*, 15 (1):4808, 2024.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. VMamba: Visual state space model. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103031–103063, 2024.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, pp. 9992–10002, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11966–11976, 2022.

Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. EdgeNeXt: Efficiently amalgamated CNN-transformer architecture for mobile vision applications. In *Computer Vision – ECCV 2022 Workshops*, Lecture Notes in Computer Science, pp. 3–20. Springer, 2023.

Omid Nejati Manzari, Hamid Ahmadabadi, Hossein Kashiani, Shahriar B. Shokouhi, and Ahmad Ayatollahi. Medvit: A robust vision transformer for generalized medical image classification. *Comput. Biol. Med.*, 157(C), May 2023. ISSN 0010-4825.

Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

Omid Nejati Manzari, Hojat Asgariandehkordi, Taha Koleilat, Yiming Xiao, and Hassan Rivaz. Medical image classification with KAN-integrated transformers and dilated neighborhood attention. *Applied Soft Computing*, 186:114045, 2026.

Alexander Noll, Robert Salzmann, and Mario V. Wüthrich. Case study: French motor third-party liability claims. *SSRN Electronic Journal*, mar 2020. doi: 10.2139/ssrn.3164764.

Yuhan Pan, Yanan Sun, and Wei Gong. Revisiting long-tailed learning: Insights from an architectural perspective. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, pp. 2294–2304, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720406.

Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, pp. 164–169, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350020.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4334–4343, 10–15 Jul 2018.

Dvir Samuel and Gal Chechik. Distributional robustness loss for long-tail learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9495–9504, October 2021.

Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pp. 1–9, 2008a.

Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pp. 1–9, 2008b. doi: 10.1109/PHM.2008.4711414.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Mingxing Tan and Quoc Le. EfficientNetV2: Smaller models and faster training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10096–10106. PMLR, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 2021a.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–42, 2021b.

Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. MobileOne: An improved one millisecond mobile backbone. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7907–7917, 2023.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12077–12090. Curran Associates, Inc., 2021.

Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9961–9970, 2021.

Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. MetaFormer baselines for vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):896–912, 2024.

Yubiao Yue and Zhenzhang Li. MedMamba: Vision mamba for medical image classification. *arXiv preprint arXiv:2403.03849*, 2024. URL https://arxiv.org/abs/2403.03849.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10795–10816, 2023.

Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan Ö. Arik, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3):3417–3425, 2022.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 1–9, Lille, France, 07–09 Jul 2015.

Table 8: Dataset statistics for ISIC 2018 and Kvasir.

Dataset	Class	Train	Val	Test	Total
ISIC 2018	NV	3807	761	3046	7614
	BKL	658	131	527	1316
	MEL	642	128	514	1284
	BCC	303	60	244	607
	AKIEC	185	37	148	370
	VASC	88	17	72	177
	DF	79	15	65	159
	Total	5762	1149	4616	11527
Kvasir	Z-line	250	50	200	500
	Pylorus	250	50	200	500
	Cecum	250	50	200	500
	Esophagitis	250	50	200	500
	Polyps	250	50	200	500
	Ulcerative Colitis	250	50	200	500
	Dyed and Lifted Polyps	250	50	200	500
	Dyed Resection Margins	250	50	200	500
	Total	2000	400	1600	4000

Table 9: Statistics of the artificially constructed imbalanced ImageNet subset.

ID	Class	Train	Val	Test	Total	Ratio
n01440764	Tench	50,000	5,000	5,000	60,000	98%
n01943899	Conch	500	50	50	600	1%
n02105056	Labrador retriever	500	50	50	600	1%
Total	–	51,000	5,100	5,100	61,200	–

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5122–5130, 2017.

A Appendix

A.1 Dataset Details

This section details the statistical summaries of the datasets employed in our experimental evaluation.

A.1.1 Medical Image Classification

For medical image classification, we utilize the ISIC 2018 Task 3 dataset for skin lesion classification and the Kvasir dataset for polyp classification. The detailed statistics are provided in Table 8.

A.1.2 Artificially Constructed Imbalanced ImageNet Subset

To evaluate the performance of our proposed REAP and baselines on tasks with extreme class imbalance, we construct an artificially imbalanced ImageNet subset. Specifically, for this subset, we augment one class using AutoAugment while leaving the other two unchanged, and then select samples to construct a severely imbalanced subset with a class ratio of 100:1:1. The statistics of this dataset are summarized in Table 9.

Table 10: Dataset statistics for image segmentation tasks.

Property	ADE20K	Kvasir-SEG
Task	Multi-class semantic segmentation	Binary segmentation (polyp)
Number of classes	150 + 1 (background)	2 (foreground/background)
Total samples	22,210	1,000
Image format	RGB, JPEG	RGB, JPEG
Annotation format	Grayscale (0–150 indices)	RGB (0/255 binary mask)
Unified resolution	No	No
Width (min/max/med)	130 / 2100 / 512	332 / 1920 / 621
Height (min/max/med)	96 / 2100 / 480	352 / 1072 / 530
Main width interval	47.9% $\in$ [512, 768)	94.1% $\in$ [512, 768)
Main height interval	46.8% $\in$ [512, 768)	90.6% $\in$ [512, 768)
Ratio $>$ 1000 px	$\sim$0.3%	3.0%

Table 11: Statistics of regression datasets.

Property	C-MAPSS FD001	C-MAPSS FD004	freMTPL2
Domain	Turbofan engine degradation	Turbofan engine degradation	Auto insurance claims
Task	RUL regression	RUL regression	Severity regression
Operating conditions	1	6	– (Tabular data)
Failure modes	1	2	–
Raw features	21 sensors + 3 settings	21 sensors + 3 settings	6 num + 4 cat
Feature dimension	14 sensors (processed)	14 sensors (processed)	6 num + 41 cat one-hot
Input format	$(N, 42, 14, 1)$ window	$(N, 42, 14, 1)$ window	$(N, 47)$ tabular
Window size	40 cycles (+2)	40 cycles (+2)	–
Train engines / policies	100	249	406,807
Train raw cycles	20,631	61,249	–
Train windows	13,702	41,131	–
Val windows	3,029	10,407	101,702
Test engines / policies	100	248	169,504
Test raw cycles	13,096	41,214	–
Total samples	16,831 (windows)	51,786 (windows)	678,013 (policies)
RUL / Severity range	Original $[7, 362]$ cycles	Original $[6, 543]$ cycles	$[1, 200, 000]$ EUR
Normalized range	$[0, 1]$ (clip@125, $\div 125$)	$[0, 1]$ (clip@125, $\div 125$)	Original value
Test RUL mean	75.5 cycles	86.6 cycles	2,084 EUR (positive)
Mean engine life	206.3 cycles	246.0 cycles	–
Data sparsity	Dense time-series	Dense time-series	96.3% zero claims
Positive claims / valid	13,702 windows	41,131 windows	24,944 (3.7%)
Severity median	–	–	1,172 EUR
Severity mean	–	–	2,081 EUR

A.1.3 Semantic Segmentation and Scene Parsing

For semantic segmentation and scene parsing tasks, we select the polyp segmentation task on the medical Kvasir-SEG dataset and the scene parsing task on the general-domain ADE20K dataset. Detailed dataset statistics are provided in Table 10.

A.1.4 Regression Tasks

We conduct regression tasks on the NASA C-MAPSS datasets (FD001 and FD004) for turbofan engine Remaining Useful Life (RUL) prediction, and the freMTPL2 dataset for auto insurance claims. Detailed dataset statistics are provided in Table 11.

A.2 Training Details

To ensure the reproducibility of REAP, this section provides comprehensive implementation details, evaluation metrics, and the baseline models used for comparison in medical image classification. All experiments described below are conducted during post-training.

A.2.1 Implementation Details

Image Classification. We evaluate on ISIC 2018, Kvasir, and an artificially constructed imbalanced ImageNet subset. All experiments adopt Segformer MiT-B4 as the backbone encoder with four REAP adapters inserted after each transformer stage. The REAP bottleneck dimensions are set to $(8, 16, 32, 32)$ with an expansion dimension of 2048 across all tasks.

Table 12 summarizes the task-specific hyperparameters. All models are optimized with AdamW and fine-tuned from a pretrained checkpoint using a three-group learning rate policy (backbone / decoder / adapter). We employ the IS_FULL_MIX training mode, where $\mathcal{K}$ key samples are selected by gradient norm and mixed with random batches at ratio r.

Table 12: Core hyperparameters for image classification experiments.

Hyperparameter	**ISIC 2018**	**Kvasir**	**ImageNet-Sub**
Number of classes	7	8	1000
Epochs	500	500	180
Batch size	16	32	32
Backbone lr	5e-6	5e-6	2e-5
Adapter lr	5e-5	5e-5	5e-4
Decoder lr	2e-5	2e-5	2e-4
LR scheduler	Cosine	Fixed	Cosine
Weight decay	1e-4	1e-4	1e-5
Label smoothing	0.1	—	—
Random erase	0.1	—	—
Key samples $\mathcal{K}$	1000	500	5000
Mix ratio r	4	4	2
IS batch size	4	8	8

Semantic Segmentation and Scene Parsing. We evaluate REAP on ADE20K and Kvasir-SEG. Both experiments use Segformer MiT-B4 as the encoder backbone with REAP Adapters inserted after each transformer stage, optimized with AdamW, and trained under the IS_FULL_MIX mode. Core hyperparameters are summarized in Table 13.

For ADE20K, we adopt a crop-based training strategy with both training and validation crop sizes set to 256×256. Four REAP Adapters with asymmetric bottleneck dimensions $(8, 16, 32, 32)$ and expansion dimension 256 are inserted after stages 1–4. For Kvasir-SEG, we employ a single bottleneck dimension of 8 with no expansion layer. The importance sampling key set size $\mathcal{K}$ and mix ratio r are adjusted according to dataset scale.

Regression Tasks. We evaluate on two regression benchmarks: C-MAPSS (sub-datasets FD001 and FD004) and freMTPL2. All regression experiments follow a two-phase training protocol: (1) Monte Carlo uniform pretraining for 100 epochs to obtain a base checkpoint, followed by (2) top-gradient key sample fine-tuning, where the top $f\%$ of training samples ranked by per-sample gradient norm are selected and used to update the model. All models adopt Segformer MiT-B0 adapted for 1D sequential (C-MAPSS) or tabular (freMTPL2) input with a sliding window of size 40. Core hyperparameters are summarized in Table 14.

For C-MAPSS, we apply the Adam optimizer and a StepLR scheduler that decays the learning rate by 0.9 every 10 steps. The effective learning rate is controlled via a scaling factor applied to the restored checkpoint learning rate ($\approx 3.14 \times 10^{-5}$). FD004, which contains 6 operating conditions and $\sim 3\times$ more training data than FD001, requires more optimization steps and a lower learning rate. The key set is rescored dynamically every 2 steps.

Table 13: Core hyperparameters for semantic segmentation and scene parsing experiments.

Hyperparameter	ADE20K	Kvasir-SEG
Number of classes	150	2 (binary)
Epochs	300	700
Batch size	8	16
Backbone lr	5e-7	1e-5
Adapter lr	5e-5	1e-5
Decoder lr	2e-5	1e-5
LR scheduler	Cosine	Cosine
Min lr	1e-7	1e-7
Weight decay	1e-4	1e-4
Crop size	256	—
Bottleneck dims	$(8, 16, 32, 32)$	8
IS batch size	4	4

For freMTPL2 severity, we employ REAP Adapters with a symmetric bottleneck of dimension 8 (4 adapters, 2 blocks each, no expansion layer) and use a three-group learning rate policy (encoder / adapter / decoder). The key set is rescored every step. The loss function is the square root of MSE computed on the raw severity scale.

Table 14: Core hyperparameters for regression experiments.

Hyperparameter	C-MAPSS FD001	C-MAPSS FD004	freMTPL2
Training samples	∼13,700	∼41,100	∼14,900 (positive)
Pretrain epochs	100	100	100
Fine-tune max steps	30	200	100
Key fraction f	1%	1%	2%
Key set size	∼137	∼411	∼298
Optimizer	Adam	Adam	Adam
Encoder lr	$\sim 1 \times 10^{-6}$	$\sim 1 \times 10^{-7}$	1×10^{-6}
Adapter lr	$\sim 1 \times 10^{-6}$	$\sim 1 \times 10^{-7}$	5×10^{-6}
Decoder lr	$\sim 1 \times 10^{-6}$	$\sim 1 \times 10^{-7}$	5×10^{-6}
LR scheduler	StepLR(10, $\gamma = 0.9$)	StepLR(10, $\gamma = 0.9$)	StepLR(10, $\gamma = 0.9$)
Rescore interval	2 steps	2 steps	1 step
Grad batch size	256	256	512 (score) / 2048 (update)

A.2.2 Evaluation Metrics

In addition to the metrics commonly used in classification, segmentation, and regression tasks, we introduce several metrics that better reflect the performance on tail-end data. The definitions of these metrics are presented in this section.

ADE20K Scene Parsing Task. We stratify the 150 categories by training pixel ratio f_c. Categories are partitioned into *head* ($f_c \geq 0.01$), *medium* ($0.001 \leq f_c < 0.01$), and *tail* ($f_c < 0.001$) classes.

Let $\text{IoU}_c = \text{TP}_c/(\text{TP}_c + \text{FP}_c + \text{FN}_c)$ denote the per-class IoU. We report stratum-specific IoUs as the mean over each partition:

$$\text{IoU}_{\text{stratum}} = \frac{1}{|\mathcal{C}_{\text{stratum}}|} \sum_{c \in \mathcal{C}_{\text{stratum}}} \text{IoU}_c, \tag{6}$$

where stratum $\in \{\text{head}, \text{medium}, \text{tail}\}$. Additionally, we define the Head-Tail Gap:

$$\text{H-T Gap} = \text{IoU}_{\text{head}} - \text{IoU}_{\text{tail}}, \tag{7}$$

which quantifies the performance disparity between frequent and rare categories. A smaller gap indicates more effective long-tail mitigation.

Kvasir-SEG Segmentation. To evaluate tail data performance on the Kvasir-SEG dataset, we use the *Rare Dice* metric. We compute the per-sample gradient norm for each validation image and rank the samples accordingly. Rare Dice is defined as the mean per-sample Dice coefficient over the top 25% of samples with the highest gradient norms.

C-MAPSS RUL Prediction. For C-MAPSS RUL prediction, we employ two domain-specific metrics. First, the PHM08 Score is an asymmetric function penalizing late predictions ($\hat{y}_i > y_i$) more severely than early warnings ($\hat{y}_i \leq y_i$), reflecting industrial costs:

$$S = \sum_{i=1}^{N} s_i, \quad s_i = \begin{cases} e^{-d_i/13} - 1, & \text{if } d_i < 0 \\ e^{d_i/10} - 1, & \text{if } d_i \geq 0 \end{cases} \tag{8}$$

where $d_i = y_i - \hat{y}_i$. Lower scores indicate better performance.

Second, we compute LOW RUL RMSE, evaluating accuracy exclusively on samples approaching failure ($y_i < 20$):

$$\text{LOW RUL RMSE} = \sqrt{\frac{1}{|\mathcal{L}|} \sum_{i \in \mathcal{L}} (\hat{y}_i - y_i)^2}, \quad \mathcal{L} = \{i \mid y_i < 20\}. \tag{9}$$

This metric specifically assesses predictive precision during the critical end-of-life phase, measuring the model's utility for immediate maintenance decisions.

freMTPL2 Regression. Following actuarial conventions, we evaluate the severity branch on policies with positive claims ($N_i > 0$). Ground-truth severity is defined as $y_i = A_i/N_i$, weighted by $w_i = N_i$, where N_i and A_i denote the claim count and total claim amount, respectively. We report:
(i) *Weighted RMSE, MSE, and MAE*, computed as $\sqrt{\sum_i w_i(\hat{y}_i - y_i)^2 / \sum_i w_i}$, $\sum_i w_i(\hat{y}_i - y_i)^2 / \sum_i w_i$, and $\sum_i w_i|\hat{y}_i - y_i| / \sum_i w_i$, respectively;
(ii) *Weighted MSE in* $\log 1p$ *space*, the core optimization objective for the severity branch:

$$\text{MSE}_{\log 1p} = \frac{\sum_i w_i \left(\log(1 + \hat{y}_i) - \log(1 + y_i)\right)^2}{\sum_i w_i}, \tag{10}$$

which stabilizes fitting by reducing sensitivity to the heavily right-skewed tail.

A.2.3 Baselines and Comparisons

To comprehensively evaluate the performance of REAP, we selected a diverse set of representative baseline models across both the **ISIC 2018** and **Kvasir** datasets. These baselines span various architectural paradigms, including classical Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), hybrid CNN-Transformer architectures, State Space Models (Mamba variants), and medical domain-specific networks.

Baseline Models on ISIC 2018. The models benchmarked on the ISIC 2018 dataset include:

- **CNN & Lightweight Architectures:** RepVGG-A1 (Ding et al., 2021), MobileOne-S4 (Vasu et al., 2023), MobileViT-v2 (Mehta & Rastegari, 2022)
- **Vision Transformers & Variants:** Swin-T (Liu et al., 2021), ConvNeXt-T (Liu et al., 2022), EdgeNeXt-B (Maaz et al., 2023), NesT-T (Zhang et al., 2022), CaiT (Touvron et al., 2021b), VMamba-T (Liu et al., 2024)
- **Medical-Specific & Specialized Networks:** HiFuse-T (Huo et al., 2024), MedMamba-T (Yue & Li, 2024), MedViT-V1-T (Manzari et al., 2023), MedViT-V2-T (Nejati Manzari et al., 2026), SegFormer (Xie et al., 2021)

Baseline Models on Kvasir. The models benchmarked on the Kvasir dataset include:

- **General Vision & Transformer Architectures:** Swin-S, ConvNeXt-S, TNT-S (Han et al., 2021), CaFormer-S (Yu et al., 2024), DeiT-S (Touvron et al., 2021a), EfficientNetV2-S (Tan & Le, 2021), CoAt-S (Xu et al., 2021), VMamba-S
- **Medical-Specific & Specialized Networks:** HiFuse-S, MedMamba-S, MedViT-V1-S, MedViT-V2-S, SegFormer

A.3 Visualization and Comparison

We provide visualization of heatmaps generated by the baseline and REAP on imaging tasks, alongside convergence behavior—specifically, the regression performance curves on C-MAPSS and the training trajectory on freMTPL2.

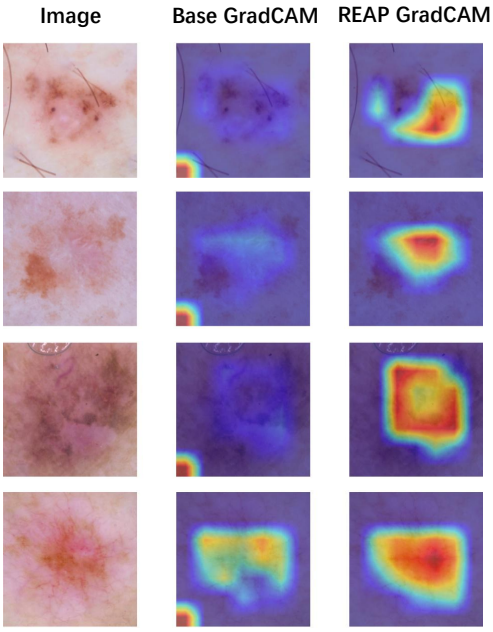

Figure 3: Heatmap visualization comparison on ISIC. The Baseline model predominantly focuses on a corner of the image, whereas REAP accurately highlights the relevant lesion regions.

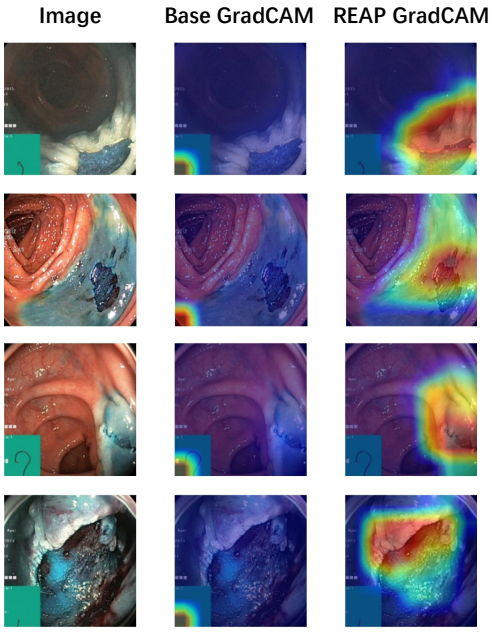

Figure 4: Heatmap visualization comparison on Kvasir. The Baseline model predominantly focuses on a corner of the image, whereas REAP accurately highlights the relevant lesion regions.

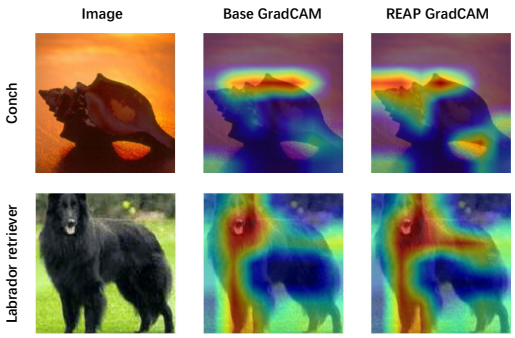

Figure 5: Heatmap visualization comparison on minority classes of the ImageNet subset. Compared to the Baseline model, REAP effectively captures multi-region semantic features across the target objects (e.g., multiple regions of the conch shell and the entire dog body instead of just the front half).

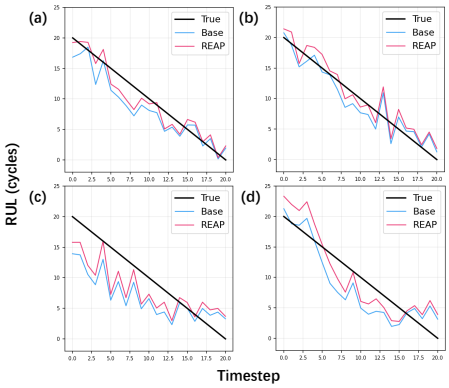

Figure 6: Comparison of RUL regression over the final 20 steps on C-MAPSS. The black line denotes ground truth, blue denotes Baseline, and red denotes REAP. REAP achieves closer alignment with the ground truth.

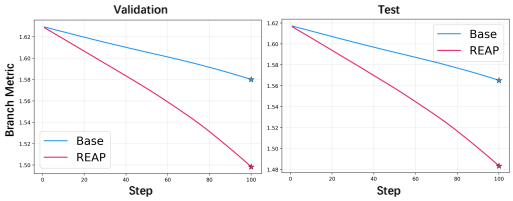

Figure 7: Training curves during post-training on the freMTPL2 dataset. Compared to the Baseline (blue), REAP (red) achieves significantly faster convergence and superior optimization performance across training steps.