28TH INT'L CONFERENCE ON PATTERN RECOGNITION · LYON 2026

Adapt-PEFT

Adaptive Parameter-Efficient Fine-Tuning for Underwater Image Enhancement

Sameer Malik · **Niki Martinel**

Machine Learning and Perception Lab , Università degli Studi di Udine, Italy

0.20%
of model parameters tuned

0.8639
SSIM — best on LSUI-L400

78–485×
fewer params than SOTA UIE

MOTIVATION

Underwater imaging matters , but the water fights back

Where it is used

1. **Marine biology**
   monitoring reefs & species
2. **Ecosystem monitoring**
   long-term ocean health
3. **Underwater archaeology**
   surveying submerged sites
4. **Autonomous exploration**
   robotic navigation & mapping

Why it is hard

Light is absorbed and scattered as it travels through water.

- **Color shifts** wavelength-dependent attenuation
- **Low contrast & haze** backscatter and turbidity
- **Lost structural detail** blur and signal loss

Degradation varies with depth, location and camera ; one fixed model cannot generalize.

THE PROBLEM

Adapting to each new environment is expensive

Today's deep UIE models achieve great quality…

GAN-based

adversarial enhancement

Diffusion

iterative generation

VAE-based

latent reconstruction

…but each new domain needs FULL retraining / fine-tuning

Large domain-specific datasets · heavy compute · long training time · slow to deploy across heterogeneous underwater conditions

Full Fine-Tuning

updates 100% of weights

Data hungry

needs large paired dataset per domain

Not scalable

infeasible for heterogeneous conditions

PEFT is the answer , but current PEFT is one-size-fits-all

Parameter-Efficient Fine-Tuning (PEFT)

Freeze the pre-trained model; train only a tiny set of extra / selected parameters.

< 1% of parameters can match full fine-tuning

The gap we target

Existing PEFT applies ONE technique uniformly to every layer.

Yet encoder and decoder layers capture very different semantics , so one fixed strategy is sub-optimal.

ENCODER layers

extract degradation-aware features , color cast, turbidity patterns

DECODER layers

reconstruct spatial detail , edges, textures, pixel-level refinement

First work to bring PEFT to underwater image enhancement with layer-specific strategy selection.

OUR IDEA

Let the network choose its own adaptation strategy , per layer

Instead of committing to one PEFT method, Adapt-PEFT dynamically MIXES two experts at selective layer via a learnable router.

SEA

Squeeze-and-Excitation Adapter

channel recalibration (additive)

SSF

Scale & Shift Features

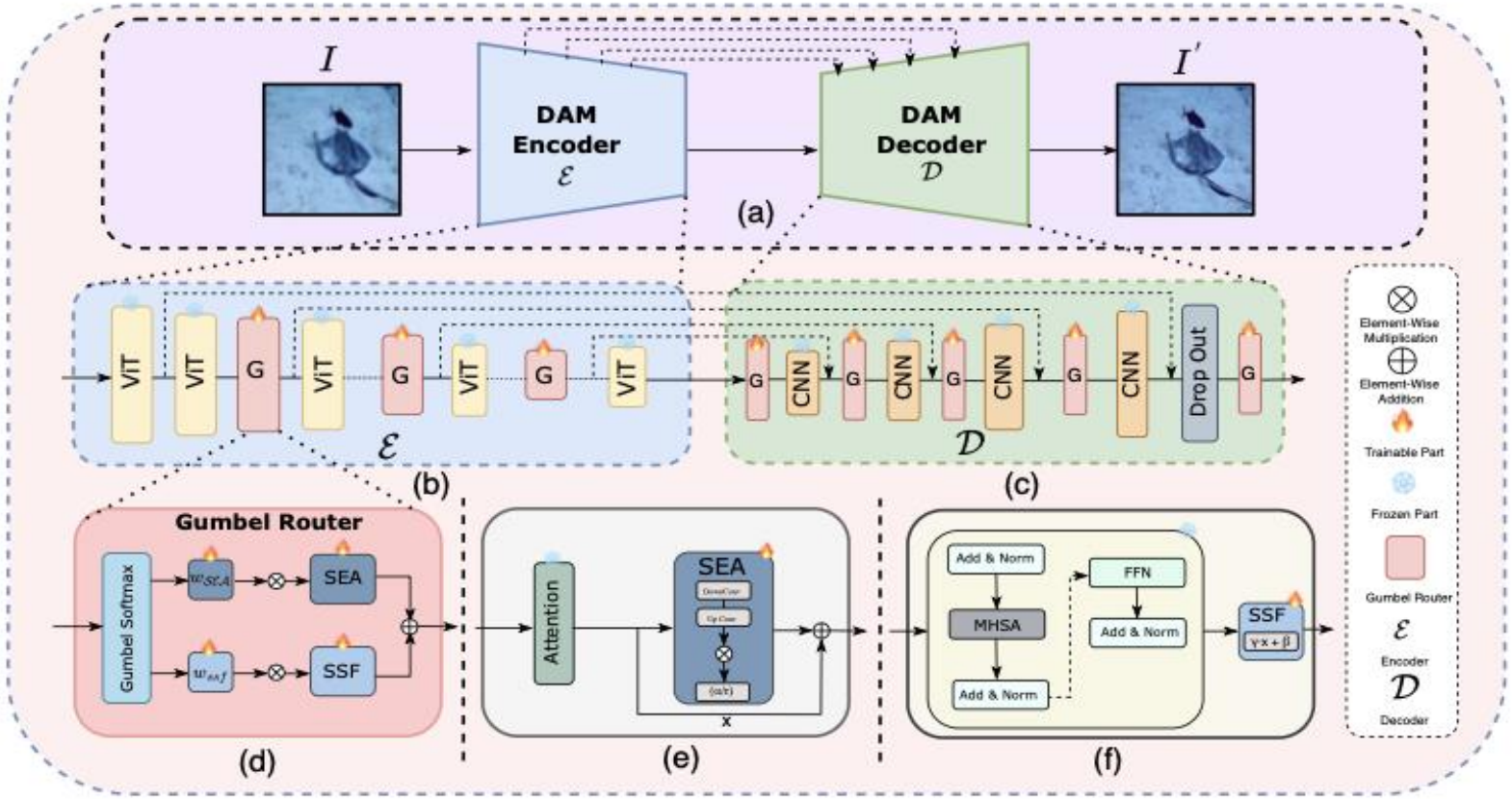
lightweight affine transform

G

Gumbel-Softmax Router

picks SEA or SSF, differentiably

ViT encoder for context · CNN decoder for detail



Dynamic Adaptive Module (DAM): at each routed point, **G** selects SEA or SSF ; global context is preserved while underwater-specific detail is restored.

THE TWO EXPERTS

Two lightweight adaptation modules

SEA

Squeeze-and-Excitation Adapter

$$SEA(x) = x + (\alpha/r) \cdot f_{excite}(\, f_{squeeze}(x)\,)$$

- Bottleneck: squeeze to rank r, then excite back
- Additive residual → stable gradients & training
- LoRA-style α/r scaling normalizes the update

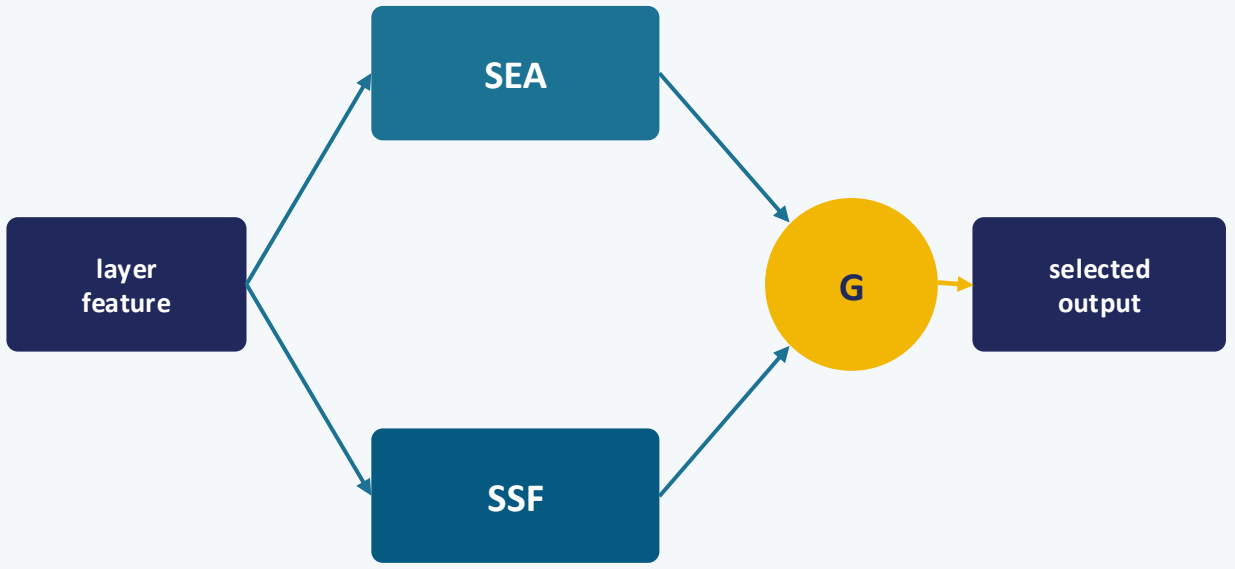
SSF

Scale & Shift Features

$$SSF(X) = \gamma \odot X + \beta$$

- Learnable affine: per-channel scale γ and shift β
- Modulates feature distributions directly
- Extremely few parameters , near-zero overhead

Gumbel-Softmax: a differentiable choice



SEA = Squeeze-and-Excitation Adapter · *SSF = Scale & Shift Features*

How the router decides

1. At each routed point, encoder layers and decoder stage two candidate adapters are available: SEA and SSF.
2. A lightweight router scores both candidates per layer, using the Gumbel-Softmax trick for a differentiable, near-discrete choice.
3. Only the winning adapter runs at inference, sparse compute, full gradient flow during training.

PUTTING IT TOGETHER

Where routing happens & how we train

Routing placement

G **Encoder**
routers at layers {3, 6, 9}; other layers frozen

G **Decoder**
an independent router at every upsampling stage

G **Final layer**
an extra adaptive router produces the output image

Training objective

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{L1}} + \mathcal{L}_{\text{Laplacian}}$$

Pixel L1 for fidelity + Laplacian-pyramid loss for multi-scale structure (sharper edges, stable training).

Setup

- 12-layer ViT · SEA rank r = 4, α = 1.0 · τ = 1.0
- AdamW, lr 1e-4 · 500 epochs · batch 8 · 256×256
- Two-stage: pre-train on LSUI-T, then PEFT-adapt (< 1% params)

EVALUATION

Benchmarks & metrics

Datasets

Full-reference (paired): LSUI-L400 · UIEB-T90

Non-reference (real, unpaired): Challenging-60 · UCCS · U45 · EUVP-330 · SQUID

Adaptation tested cross-domain: pre-train on LSUI-T, fine-tune on UIEB , no target-specific training.

Metrics

Paired: PSNR · SSIM · LPIPS

Unpaired: UCIQE · UIQM

Higher PSNR/SSIM/UCIQE/UIQM better; lower LPIPS better.

0.172 M

trainable params

0.20 %

of the 87.9 M model

7

datasets evaluated

SOTA quality at a fraction of the cost

0.8639

Best SSIM on LSUI-L400

matches full fine-tuning (0.8625)

78–485×

Fewer trainable params

than UShape-T, Spectroformer, CE-VAE…

PSNR, SSIM & parameter budget

Method	LSUI			UIEB			#Param [M]↓
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	
UDCP[6]	13.85	0.6009	0.35	11.61	0.5530	0.28	–
UIBLA[32]	18.11	0.7445	0.32	15.72	0.6831	0.26	–
UGAN[9]	19.78	0.8022	0.34	16.16	0.7259	0.40	–
Cluie-Net[24]	18.88	0.7953	0.30	20.02	0.8627	0.19	13.40
TWIN[28]	20.11	0.8076	0.31	22.04	0.8253	0.23	11.38
UShape-Transformer[31]	23.64	0.8419	0.27	20.91	0.8037	0.23	31.59
Spectroformer†[19]	20.41	0.8127	0.30	**25.79**	**0.9189**	**0.12**	2.43
CE-VAE‡[33]	**25.32**	**0.8637**	**0.23**	19.24	0.7848	0.27	83.44
*Ours-LSUI-T	**24.27**	**0.8639**	**0.26**	20.54	0.8567	0.20	**0.172**
*Ours-UIEB	21.45	0.8385	**0.26**	**22.49**	**0.8910**	**0.17**	

Best SSIM on LSUI-L400 with the smallest budget by far. Spectroformer/CE-VAE were trained directly on their target set, ours is pure cross-domain adaptation.

Adaptive routing beats uniform PEFT and FFT

PEFT methods (same backbone)

Method	% Params↓	LSUI				UIEB			
		PSNR↑	SSIM↑	UCIQE↑	UIQM↑	PSNR↑	SSIM↑	UCIQE↑	UIQM↑
No-FT	–	24.23	0.8627	0.5815	3.0209	20.05	0.8484	0.5913	3.0762
FULL-FT	100%	24.04	0.8625	**0.5818**	3.0563	**23.26**	**0.9076**	**0.6112**	3.1776
LayerNorm	0.04%	**24.28**	**0.8631**	0.5815	3.0331	21.75	0.8762	0.5986	3.1602
BitFit	0.06%	24.26	0.8625	0.5810	3.0386	21.10	0.8591	0.5887	3.1830
LoRA	0.17%	**24.27**	0.8629	0.5813	3.0279	21.05	0.8633	0.5944	3.1417
Adapters	1.99%	19.59	0.7827	0.5747	**3.1649**	16.94	0.7698	0.5935	3.1277
SSF	0.23%	23.53	0.8558	0.5817	3.0739	21.87	0.8716	0.6015	**3.2324**
VPT	0.03%	20.69	0.8041	**0.5857**	**3.1933**	20.40	0.8239	0.5989	**3.3239**
Ours	0.20%	**24.27**	**0.8639**	0.5817	3.0453	**22.49**	**0.8910**	**0.6037**	3.1294

Best SSIM on LSUI; 2nd-best on UIEB , above every uniform PEFT baseline.

Beats full fine-tuning

On LSUI-T our SSIM ties FFT and PSNR exceeds it (24.27 vs 24.04 dB) , while updating **485× fewer parameters.**

Routing > static selection

Ablation: replacing the learned router with a fixed SEA-encoder / SSF-decoder assignment **drops PSNR by 0.23 / 0.14 dB** (LSUI / UIEB). Dynamic, per-layer choice is what wins.

What to take away

1

Adaptive PEFT selection

A framework that dynamically chooses between existing PEFT methods (SEA / SSF) layer-specific, not one fixed strategy.

2

Dynamic Adaptive Module

Gumbel-Softmax routing adds negligible parameters yet delivers superior cross-domain adaptation.

3

First PEFT for UIE

State-of-the-art results while tuning < 1% of parameters, beating uniform PEFT and full fine-tuning.

Thank you

Adapt-PEFT : Adaptive, layer-specific PEFT for underwater image enhancement.

- State-of-the-art SSIM 0.8639 — while tuning only 0.20% of parameters.

Sameer Malik · **Niki Martinel** malik.sameer@spes.uniud.it

Machine Learning and Perception Lab , University of Udine, Italy — Questions welcome