Seeing Through Seismic Noise with Soft Spatial Blending

Parameter-Efficient Soft Spatial Blending of Vision Foundation Models for Seismic Denoising

Alexis FIMEYER[1]

Supervisors: Dr. Jing Sun[1], Dr. Tiexing Wang[3], Dr. Eric Verschuur[2], Jiahua Zhao[2,4]

[1]EEMCS, Delft University of Technology, The Netherlands
[2]Faculty of Civil Engineering and Geosciences, Delft University of Technology, The Netherlands
[3]R&D Department, Shearwater GeoServices, UK
[4]The Cyprus Institute, Cyprus

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Alexis FIMEYER
Final project course: CSE3000 Research Project
Thesis committee: Dr. Jing Sun[1], Dr. Tiexing Wang[3], Dr. Eric Verschuur[2], Jiahua Zhao [2,4]
Examiner: Dr. Petr Kellnhofer[1]

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.
The accompanying code repository is available at `https://github.com/AlexisFimeyer/research-project`.

Abstract

Active seismic imaging is used to infer subsurface structure from reflected wavefields, but acquisition and ambient noise can obscure weak reflectors and reduce interpretation reliability. Seismic denoising must remove noise while keeping geological structure intact. This thesis studies a parameter-efficient method to adapt pretrained vision foundation models to this task. The method treats each seismic section as a 2D grayscale image, maps it into a format compatible with vision backbones, and applies Low-Rank Adaptation (LoRA) to limit the number of trainable parameters. It then combines the denoised outputs of multiple adapted vision models through a learned soft spatial blender. This blender merges the expert predictions at the pixel level, allowing the final model to use complementary architectural strengths such as multiscale representation and long-range dependencies. The method is evaluated against a seismic foundation model baseline, using both quantitative metrics and qualitative inspection. Across 25 seed/split repetitions, the residual joint spatial blender achieves a mean absolute error of 0.0463, a peak signal-to-noise ratio of 19.14 dB, and a structural similarity index of 0.9727, substantially outperforming the standalone adapted experts and the frozen baseline. These results show that jointly trained spatial fusion improves seismic denoising performance while keeping training parameter-efficient.

1 Introduction

In active seismic methods, controlled sources generate wavefields that are recorded by receivers after interacting with subsurface structures [1]. Active seismic imaging is used to study the Earth's subsurface in applications such as geothermal assessment, hydrocarbon exploration, carbon-storage monitoring, and groundwater mapping. Recorded waveforms contain ambient and acquisition noise, which can hide fine geological structure and make interpretation less reliable. Classical denoising methods address this problem with structure-aware transforms, curvelets, reduced-rank filtering, and sparse representations [2], [3], [4], [5]. The difficulty is that denoising must suppress noise without distorting amplitude, phase, or the continuity of geologic reflectors.

Deep convolutional models, including fully convolutional networks, U-Net-style encoder-decoders, and residual denoisers, are widely used for dense restoration tasks because they learn image-to-image mappings from noisy to clean data [6], [7], [8], [9]. More recent work has introduced *seismic foundation models* (SFMs), which are trained on large seismic datasets and then fine-tuned for downstream tasks [10]. While promising, these models require large amounts of data, substantial computing power, and significant engineering efforts.

A cost-effective alternative is to reuse existing *vision foundation models* (VFMs) trained on large image collections, an idea recently explored for active and passive seismic denoising through parameter-efficient VFM adaptation [11]. The approach draws on attention-based sequence modeling, vision transformers, self-supervised visual representation learning, masked autoencoding, and large reusable visual models [12], [13], [14], [15], [16], [17]. Backbones such as DINOv3 [18] and SwinV2 [19] can provide transferable features, but applying them to seismic data introduces three challenges. First, VFMs expect three-channel RGB input, whereas seismic sections are generally single-channel. Second, full fine-tuning is computationally expensive and can overfit when training pairs are limited. Third, a single backbone may not use the complementary strengths of different architectures: SwinV2 provides hierarchical multiscale representations, while DINOv3 is more suited for long-range dependencies.

Parameter-efficient methods, including adapters and LoRA, reduce the cost of adaptation by training only small added components while keeping most pretrained weights frozen [20], [21], [22]. A small input stem can handle the channel mismatch by mapping single-channel seismic data to a three-channel representation. The remaining question is how to use more than one adapted backbone. This thesis uses a mixture-of-experts (MoE) design in which multiple LoRA-adapted backbones are fused by a learned spatial router. The router blends expert outputs at each pixel rather than choosing a single expert for the whole image. Accordingly, the central research question is: *Can a trained soft spatial blender (SSB) that combines multiple adapted vision experts improve seismic denoising performance compared to an SFM baseline, while remaining computationally efficient?* This thesis develops, implements, and evaluates such a parameter-efficient approach.

1.1 Problem Statement

The project addresses supervised seismic denoising from paired noisy and clean 2D sections. Given a noisy seismic section $x \in \mathbb{R}^{H \times W}$, the goal is to predict a clean section $\hat{y}$ that approximates the target y while preserving coherent events and amplitude structure:

$$f_\theta : x \mapsto \hat{y}, \qquad \hat{y} \approx y. \tag{1}$$

The constraints are:

- the supervised dataset used in this thesis is small, containing 2000 paired examples;
- the model must process single-channel seismic amplitudes, although the selected backbones expect RGB images;
- training should avoid full fine tuning of large pretrained encoders;
- the experiments should be reproducible from saved configurations, checkpoints, and metric files

1.2 Research Objectives

The project is organized around four objectives:

- Design a parameter-efficient denoising architecture that uses pretrained vision backbones for single-channel seismic input.
- Train and evaluate standalone DINOv3 and SwinV2 LoRA denoisers under the same repeated supervised evaluation protocol.
- Build a learned spatial fusion model that combines both experts with a router.
- Assess whether the added complexity is justified by denoising performance, compute cost, and inference time.

1.3 Research Questions

The central research question is:

> *Can a trained soft spatial blender that combines multiple parameter-efficiently adapted vision foundation-model experts improve seismic denoising performance compared to a SFM, while remaining computationally efficient?*

This question is decomposed into four sub-questions:

RQ1. How well do DINOv3 and SwinV2 perform as standalone LoRA-adapted seismic denoisers?

RQ2. Does an expert-aware router improve over the stronger adapted expert?

RQ3. Does joint training of the router and trainable expert components improve over router-only fusion?

RQ4. What performance gains does the router approach provide, and what are the tradeoffs in inference time, compute cost, and implementation complexity?

2 Background & Related Work

2.1 Seismic Denoising

Seismic denoising differs from generic image denoising because the shape of the signal has physical meaning. Reflectors appear as mostly coherent events, and their curvature, continuity, and amplitude carry information. At the same time, important geological features can also be abrupt, for example where faults offset otherwise continuous reflectors. A denoising method should therefore preserve signal phase, event continuity, relative amplitude, and sharp structural discontinuities. Classical methods often use curvelet-domain sparsity, reduced-rank structure, and dictionary learning for this reason. This requirement motivates models that combine local edge sensitivity with wider spatial context. U-Net-style encoder-decoder models, fully convolutional dense predictors, and residual CNN denoisers are common choices for image-to-image restoration because they combine multiscale context with dense prediction. This project keeps that dense prediction structure in the decoder, but replaces a conventional learned encoder with pretrained vision backbones adapted to seismic input.

2.2 Vision Foundational Models

Vision foundation models are large pretrained visual encoders designed to transfer to many tasks. They build on attention and vision transformers, self-supervised representation learning, and masked-autoencoder-style pretraining. DINO-style self-supervised transformers learn representations without manual labels and transfer well to several visual tasks. SwinV2 uses hierarchical shifted-window attention, which allows transformer backbones to scale while preserving multi-resolution feature maps [19]. For seismic denoising, these two models are complementary: DINOv3 provides global token mixing, while SwinV2 provides hierarchical features that suit dense prediction. Transfer from natural images to seismic sections is still not direct. The input channels have different meaning, the task is dense regression rather than classification, and absolute amplitude scale can matter. In this thesis, the VFM is therefore treated as an adaptable encoder, not as a complete denoising model.

2.3 Parameter-Efficient Fine-Tuning

Full fine-tuning updates all of the pretrained encoder weights. This can be expensive and can overfit when the dataset is small. Parameter-efficient transfer methods reduce this cost by adapting only small modules, such as adapters, or low-rank updates [20], [21], [22]. LoRA [22] reduces the number of trainable parameters by adding low-rank residual updates to selected frozen linear layers. For a frozen linear transformation with weight W_0, LoRA computes:

$$h = W_0 x + \frac{\alpha}{r} BAx, \tag{2}$$

where $A \in \mathbb{R}^{r \times d}$ and $B \in \mathbb{R}^{k \times r}$ are trainable low-rank matrices, r is the rank, and α is a scaling factor. The base weight W_0 remains frozen. In this thesis, all LoRA runs use $r = 16$ and $\alpha = 64$. These values were chosen as a moderate-capacity setting: rank 16 keeps the number of trainable parameters limited, while the scaling factor 64 gives the low-rank update enough strength to adapt the frozen vision backbones.

2.4 Mixture-of-Experts Fusion

Mixture-of-experts (MoE) models combine multiple predictors through a router or gating function that learns how to weight specialized experts [23], [24], [25]. In sparse MoE systems, the router selects a small subset of experts or tokens to reduce compute [25],

[26], [27]. In this thesis, the goal is different: both experts are evaluated, and the router learns a soft spatial blend. This is appropriate for seismic denoising because different regions of the same section may benefit from different inductive biases.

The fusion equation is:

$$\hat{y}_{\mathrm{SSB}} = (1-g)\hat{y}_S + g\hat{y}_D, \tag{3}$$

where $\hat{y}_S$ is the SwinV2 expert prediction, $\hat{y}_D$ is the DINOv3 expert prediction, and $g \in [0,1]^{H\times W}$ is the spatial gate. A gate value near 0 favors SwinV2; a value near 1 favors DINOv3.

3 Methodology

The project uses an experimental approach. Each model variant is evaluated under a repeated seed/split protocol. Five random initialization seeds, 42, 43, 44, 45, and 46, are crossed with five independent dataset splits, giving 25 complete experiment repetitions:

1. Train standalone DINOv3 LoRA and SwinV2 LoRA denoisers.
2. Train a router-only SSB with frozen experts.
3. Jointly fine-tune the router and trainable experts.

This order avoids directly jumping to the final architecture and makes it possible to attribute improvements to expert quality, router training, joint router-expert training and other steps.

3.1 Dataset and Evaluation

The supervised denoising dataset used in this work is taken from the downstream signal-processing denoising data released with the Seismic Foundation Model (SFM). In the SFM work, the authors generated 2000 noise-free seismic samples and added random noise to create paired noisy-clean training examples. The released downstream denoising data are provided as 224×224 seismic samples by the SFM project [10].

For each split, the dataset was partitioned as follows during training:

- training: 1600 pairs;
- validation: 200 pairs;
- test: 200 pairs.

The same five splits are reused across model variants, and each split is paired with each of the five initialization seeds. Reported uncertainty values for trained models are therefore standard deviations across 25 seed/split repetitions.

The evaluation uses mean absolute error (MAE), mean squared error (MSE), peak signal-to-noise ratio (PSNR), and the structural similarity index (SSIM) [28]; SSIM and multi-scale SSIM (MS-SSIM) are commonly used when structural similarity is more important than pointwise error alone [29]. MAE and MSE measure amplitude error:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|, \qquad \mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2. \tag{4}$$

PSNR summarizes error relative to the signal range:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{L^2}{\mathrm{MSE}}\right), \tag{5}$$

where L is the data range used by the implementation. SSIM measures structural similarity, which is important because reflector continuity is not fully captured by previous metrics:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \tag{6}$$

Here, μ_x and μ_y are local means, σ_x^2 and σ_y^2 are local variances, σ_{xy} is the local covariance, and C_1 and C_2 are small constants for numerical stability.

3.2 System Architecture

The complete residual SSB denoising pipeline is shown in Figure 1. The noisy seismic input x is passed through two LoRA-adapted foundation-model experts. The SwinV2 expert produces $\hat{y}_S$, while the DINOv3 expert produces $\hat{y}_D$. Rather than selecting one expert globally, the model learns a dense spatial gate g that blends the two predictions at each pixel using Eq. 3.

3.3 Frequency-Aware Input Stem

The pretrained backbones expect three-channel input, while seismic sections are single-channel. The input stem builds a learned three-channel representation from the original normalized seismic image and a fixed high-pass response. The high-pass branch uses a Laplacian filter:

$$K_\Delta = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \tag{7}$$

The original image and high-pass response are concatenated:

$$z_0 = \mathrm{concat}\left(\hat{x}, K_\Delta * \hat{x}\right), \tag{8}$$

then passed through trainable convolutions, normalization, GELU activations, and a final 1×1 convolution that produces three channels.

3.4 Standalone Experts

Both standalone experts follow the same high-level computation:

$$\hat{y} = D_\theta\left(E_\phi\left(S_\psi(\hat{x})\right)\right), \tag{9}$$

where S_ψ is the input stem, E_ϕ is a pretrained encoder with LoRA adaptation, and D_θ is a dense decoder.

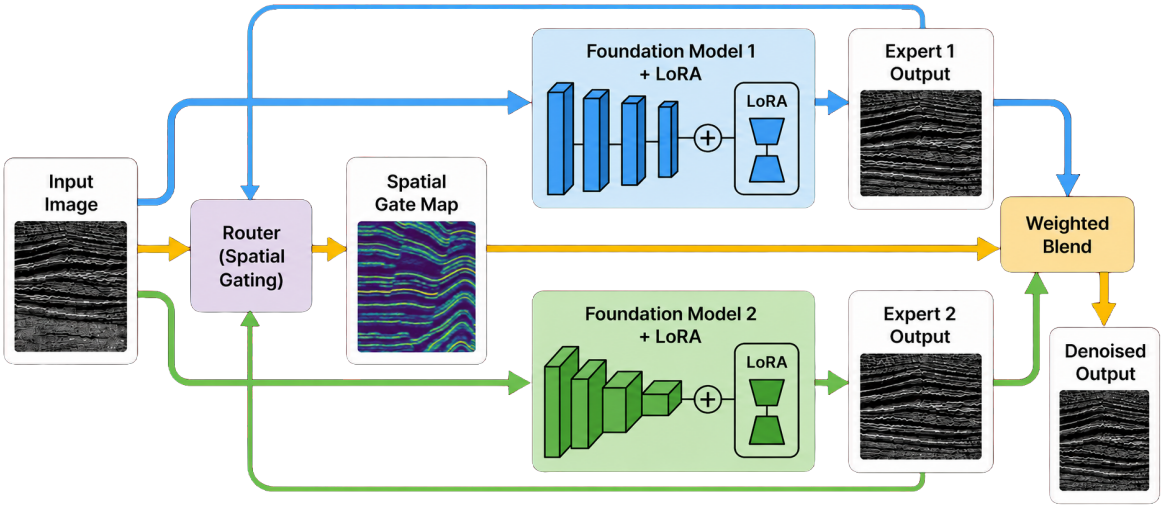

Figure 1: Overall residual SSB architecture. The noisy seismic input is processed by two LoRA-adapted foundation-model experts, producing expert predictions $\hat{y}_S$ and $\hat{y}_D$. A spatial router predicts a dense gate map g, and the final denoised output is obtained by the pixelwise weighted blend $\hat{y}_{\mathrm{SSB}} = (1-g)\hat{y}_S + g\hat{y}_D$.

The decoder follows the dense prediction principle used in fully convolutional and U-Net-style architectures, where spatial feature maps are transformed back into an image-sized output [6], [7].

Both experts use LoRA on the main transformer linear layers: `qkv` denotes the combined query, key, and value projection used in self-attention; `proj` is the output projection after attention; and `fc1` and `fc2` are the two fully connected layers in the feed-forward block. DINOv3 has decoder widths $(192, 96, 48, 24)$, and SwinV2 has decoder widths $(384, 256, 192, 96)$.

3.5 Residual Expert-Aware Router

The residual router uses the same information available at inference time:

$$r = \mathrm{concat}\left(\hat{x}, \hat{y}_S, \hat{y}_D, |\hat{y}_D - \hat{y}_S|\right). \tag{10}$$

The disagreement map $|\hat{y}_D - \hat{y}_S|$ tells the router where the experts agree and where a stronger routing decision may be needed. It is not a target residual and does not require the clean image at inference time.

The term residual refers to internal skip connections inside the router, following the residual-learning idea that a block can learn a correction to its input rather than a full replacement [8]. A residual block computes:

$$h_{\mathrm{out}} = h_{\mathrm{in}} + F(h_{\mathrm{in}}), \tag{11}$$

where F is a small stack of normalization, activation, and convolution layers. The implementation uses GroupNorm and GELU, which are standard choices for stable feature normalization and smooth nonlinear activation [30], [31]. This lets the router learn corrections to an existing feature map rather than replacing the feature map completely.

The router uses channel widths $(32, 64, 32)$. It projects the router input to local features, applies a local residual convolution block, and sends those features through a downsampled context branch. The context branch contains residual blocks, including a dilated block, so the router can use a wider spatial neighborhood without losing dense output resolution [32], [33]. The context features are upsampled and fused with the local features before a final 1×1 convolution and sigmoid produce the gate map. A diagram of the architecture can be seen in Figure 2.

The residual router is trained through the denoising objective of the final blended prediction, not through a separate ground-truth gate. For the residual router-only and residual joint SSB runs, the prediction loss is the L1 error between the blended output and the clean target:

$$\mathcal{L}_{\mathrm{rec}} = \left\|\hat{y}_{\mathrm{SSB}} - y\right\|_1. \tag{12}$$

To discourage noisy pixel-to-pixel gate changes, the training objective adds a total-variation penalty on the spatial gate [34]:

$$\begin{aligned}\mathcal{L}_{\mathrm{TV}}(g) = &\frac{1}{(H-1)W}\sum_{i=1}^{H-1}\sum_{j=1}^{W}|g_{i+1,j} - g_{i,j}| \\ &+ \frac{1}{H(W-1)}\sum_{i=1}^{H}\sum_{j=1}^{W-1}|g_{i,j+1} - g_{i,j}|\end{aligned}. \tag{13}$$

The residual SSB loss is therefore:

$$\mathcal{L}_{\mathrm{SSB}} = \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{TV}}\mathcal{L}_{\mathrm{TV}}(g). \tag{14}$$

The router-only run uses $\lambda_{\mathrm{TV}} = 0.002$, while the joint residual SSB uses $\lambda_{\mathrm{TV}} = 0.005$. A gate-balance term was used during training to avoid the router collapsing to the best expert at the start.

3.6 Training Setup

Training and evaluation were run on a single NVIDIA GeForce RTX 4090 GPU with 24 GB memory [35],

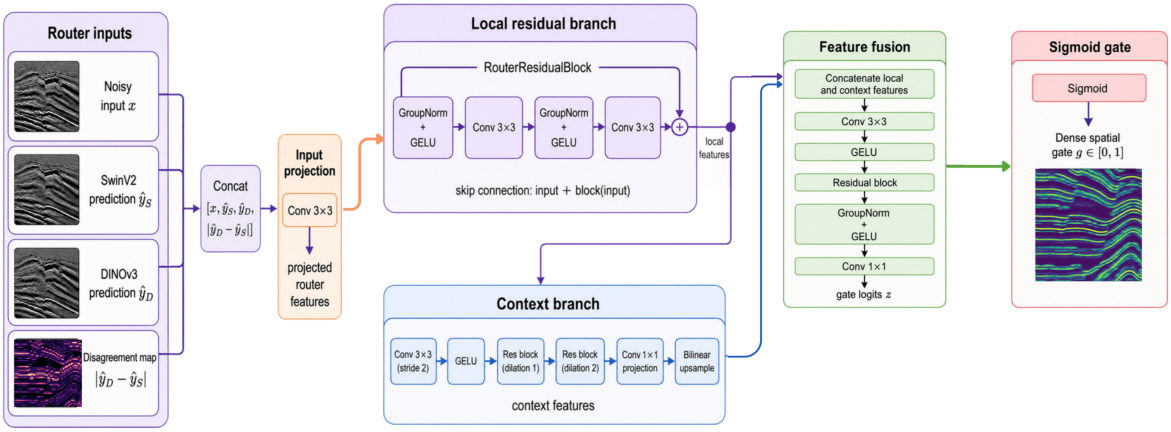

Figure 2: Residual router architecture. The router receives the noisy input x, the SwinV2 prediction $\hat{y}_S$, the DINOv3 prediction $\hat{y}_D$, and the disagreement map $|\hat{y}_D - \hat{y}_S|$. After a 3×3 input projection, local residual features and broader context features are extracted, concatenated, and mapped to gate logits. A sigmoid produces the dense spatial gate $g \in [0, 1]$, where values near 0 favor SwinV2 and values near 1 favor DINOv3.

using PyTorch [36] for model implementation and optimization. Pretrained DINOv3 and SwinV2 backbones were loaded through `timm` [37]. Mixed precision was enabled for training. No test-time augmentation was used. Each reported trained-model value is the mean over 25 runs formed by five initialization seeds and five dataset splits. Reported uncertainty values are standard deviations over these 25 repetitions; the frozen SFM-Base baseline is deterministic and is reported as a point estimate.

The final evaluation compares SFM-Base, a seismic foundation model based on masked-autoencoder-style pretraining for geophysical data [10], [16], with four adapted model variants: DINOv3 LoRA, SwinV2 LoRA, residual router-only SSB, and residual joint SSB.

4 Results

4.1 Quantitative Performance

Table 2 reports the held-out test performance, including SFM-Base as a frozen foundation-model baseline. SFM-Base has competitive PSNR among the non-fused baselines but lower SSIM, indicating weaker preservation of seismic reflector structure while having more parameters. DINOv3 LoRA and SwinV2 LoRA have similar standalone PSNR and SSIM, but DINOv3 has lower MAE and MSE. SwinV2 remains useful because the residual router can still exploit complementary local structure from its prediction.

The residual router-only SSB improves over both standalone experts, reaching MAE 0.1033, MSE 0.0263, PSNR 15.80 dB, and SSIM 0.8748 while training only 225,793 router parameters, being only $\sim 0.37\%$ of the total parameter count. This shows that spatial fusion is useful even when the experts remain frozen.

The residual joint SSB gives the best result. It reduces MAE to 0.0463, MSE to 0.0122, and increases PSNR to 19.14 dB and SSIM to 0.9727. Compared with residual router-only fusion, joint fine-tuning reduces MAE by 55.2%, reduces MSE by 53.6%, increases PSNR by 3.34 dB, and increases SSIM by 0.0979.

4.2 Inference Speed

Inference speed was measured on a NVIDIA GeForce RTX 4090 GPU with 24 GB memory on the same held-out test patches and batch size 16. The benchmark excludes data loading time by timing only the forward pass after two warm-up batches. Throughput is reported as patches per second:

$$\text{patches/s} = \frac{1}{\text{seconds/patch}}. \tag{15}$$

Table 3 shows that SFM-Base is the fastest model benchmark. The residual joint SSB is still slower than a single expert because it evaluates both experts and the residual router, but code-level inference optimization improved its throughput by a factor of 2.83, from 5.52 to 15.62 patches/s. The main changes were fusing static LoRA weights into the base dense kernels before timing, so each LoRA projection runs as one dense projection at inference, and rewriting the final blend from $g\hat{y}_D + (1-g)\hat{y}_S$ to $\hat{y}_S + g(\hat{y}_D - \hat{y}_S)$. The expert difference is also reused as the router disagreement feature, avoiding redundant element-wise computation.

4.3 Router Behavior

The residual router-only model has a gate mean of 0.6680 and assigns more than half the weight to DINOv3 for 71.3% of pixels. After joint fine-tuning, the final residual joint SSB has a lower gate mean of 0.3274 and a gate fraction over 0.5 of 23.4%. This shift does not mean DINOv3 is unused. It means the jointly trained system changes the relative roles of the two experts: DINOv3 is slightly stronger as a standalone model, while SwinV2 contributes more strongly in the final joint mixture after co-adaptation.

Table 1: Main hyperparameters for the final reported runs.

Run	Epochs	Batch	LR	Weight decay	Warmup	Loss
DINOv3 LoRA	50	16	6.0×10^{-4}	1.0×10^{-2}	100	L1
SwinV2 LoRA	50	16	8.0×10^{-4}	1.0×10^{-2}	100	L1
Residual router-only SSB	50	16	1.0×10^{-3}	1.0×10^{-4}	100	L1 + gate TV
Residual joint SSB	50	16	2.0×10^{-4}	1.0×10^{-4}	100	L1 + gate TV

Table 2: Held-out test performance for the final reported models and SFM-Base baseline, ordered from worst to best by SSIM. Metric values for trained models are reported as mean ± standard deviation over 25 seed/split repetitions. Lower is better for MAE and MSE; higher is better for PSNR and SSIM. Gate statistics are only defined for SSB models. SFM-Base is evaluated as a frozen deterministic baseline.

Model	Trainable	Total params	MAE	MSE	PSNR	SSIM	Gate mean	Gate > 0.5
SFM-Base	–	88,878,336	0.3343	0.2364	6.26	0.3463	–	–
SwinV2 LoRA	9,476,900	37,041,377	0.3310 ± 0.0160	0.3382 ± 0.0221	4.71 ± 0.42	0.4863 ± 0.0201	–	–
DINOv3 LoRA	1,717,873	23,304,817	0.2887 ± 0.0138	0.3171 ± 0.0195	4.99 ± 0.36	0.5013 ± 0.0184	–	–
Residual router-only SSB	225,793	60,571,987	0.1033 ± 0.0076	0.0263 ± 0.0050	15.80 ± 0.26	0.8748 ± 0.0096	0.6680	0.7133
Residual joint SSB	**11,420,575**	**60,571,987**	$\mathbf{0.0463 \pm 0.0046}$	$\mathbf{0.0122 \pm 0.0029}$	$\mathbf{19.14 \pm 0.20}$	$\mathbf{0.9727 \pm 0.0071}$	**0.3274**	**0.2343**

Table 3: Inference throughput on held-out test patches.

Model	s/patch	patches/s
SwinV2 LoRA expert	0.0775	12.89
DINOv3 LoRA expert	0.0526	19.01
Residual joint SSB	0.0640	15.62
SFM-Base	0.0222	44.89

Figure 3 shows held-out qualitative examples with the noisy input, residual-joint SSB gate, prediction, and residual. The gate varies spatially with seismic structure rather than selecting one expert globally.

5 Discussion

5.1 Answering the Research Questions

RQ1: standalone experts. The research question asked how well the two LoRA-adapted vision backbones perform when they are used as standalone seismic denoisers. DINOv3 LoRA reaches MAE 0.2887 ± 0.0138 and SSIM 0.5013 ± 0.0184, while SwinV2 LoRA reaches MAE 0.3310 ± 0.0160 and SSIM 0.4863 ± 0.0201. These results show that DINOv3 is the stronger individual expert overall, especially in terms of MAE, MSE, and SSIM, although the PSNR values of the two standalone experts are similar. However, the standalone results also show that adapting a single vision backbone is not sufficient to reach the performance of the residual SSB models. The main value of these experts is therefore not only their individual denoising ability, but also the complementary information they provide when they are combined by a spatial blending model.

RQ2: residual router-only fusion. The research question considered whether a residual router can improve denoising performance without further updating the adapted experts. The residual router-only SSB improves over both standalone experts while training only the router. This indicates that the router is able to use the noisy input, the two expert predictions, and their disagreement map to make a more informed spatial blend than either expert can provide alone. The improvement is important because it shows that fusion is not just averaging two outputs, but learning where each expert is more useful within the seismic section.

RQ3: residual joint fine-tuning. The research question addressed whether the router and trainable expert should be optimized together. Jointly training the residual router with the trainable parts of the experts gives the strongest result. The improvement over residual router-only fusion suggests that the experts and router benefit from co-adaptation. In other words, the final model does not only learn how to combine two fixed denoisers after training; it also allows the denoisers to adjust their representations in a way that makes them more useful for the learned spatial mixture.

RQ4: computational tradeoffs. The research question focused on the tradeoff between denoising quality and inference cost. The router-based approach provides substantial performance gains, especially after residual joint fine-tuning, but these gains come with additional inference cost. This cost is expected, because both experts must be evaluated before the router can blend their outputs. As a result, the residual joint SSB remains slower than the DINOv3 standalone expert in the reported evaluation, although the inference optimization reduced the gap substantially. At the same time, the method is still more efficient to train than full fine-tuning, since most backbone parameters remain frozen and only the LoRA modules, decoder components, and router are updated. The final method should therefore be understood as a compromise: it improves denoising quality and keeps training parameter-efficient, but it does not minimize inference time.

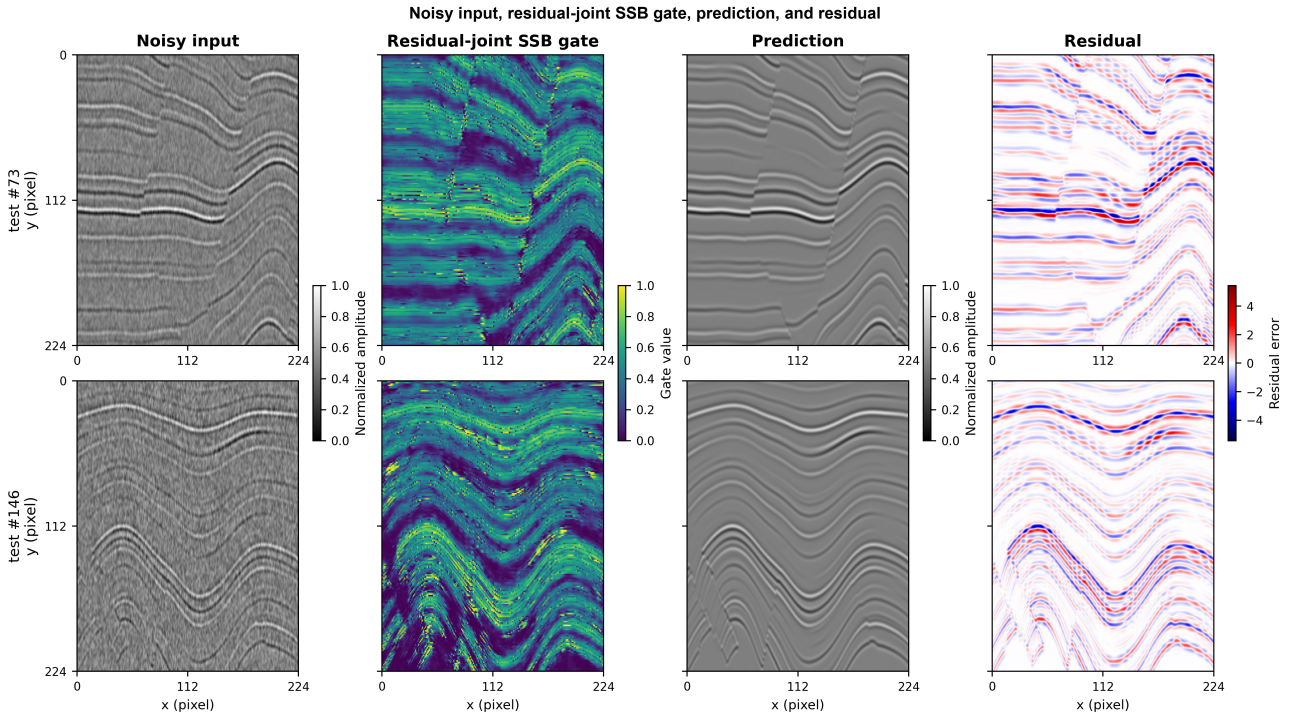

Figure 3: Qualitative examples for two held-out test samples. Shown: the noisy input, residual-joint SSB gate, prediction column, and residuals. The grayscale colorbars show normalized amplitude, the gate colorbar shows the spatial blending weights learned by the router, and the residual colorbar shows prediction error.

5.2 Implications

The results suggest that pretrained vision backbones can be useful for seismic denoising when they are adapted in a parameter-efficient way. Instead of training a large seismic model from scratch or fully fine-tuning a vision model, LoRA adaptation provides a lighter way to reuse pretrained representations for dense regression. This is relevant for seismic applications because labelled noisy-clean pairs are limited, while training or fine-tuning large models can quickly become expensive.

The main implication is that combining different adapted backbones can be more effective than selecting a single best model. DINOv3 and SwinV2 have different architectural strengths, and the spatial blender allows the final system to use them differently across the seismic section. This is useful because seismic denoising often requires both local sensitivity to small reflector details and broader contextual understanding of continuous structures. A spatially varying blend can support both requirements better than a single global model choice.

At the same time, the improvement comes with extra inference cost because both experts must be evaluated. The approach is therefore most suitable for settings where denoising quality is more important than minimal runtime, and where GPU resources are available for training and deployment. For use cases with strict real-time constraints, a single-expert model or a distilled version of the SSB may be more practical.

5.3 Limitations

The dataset contains 2000 paired examples, which is useful for controlled experiments but not enough to claim broad generalization. All reported results are based on the same supervised denoising dataset, so the conclusions mainly show that the method works under this data distribution. They do not prove that the same gains will transfer to other acquisition settings, geological regions, preprocessing pipelines, or noise types.

The clean targets are treated as ground truth, while real seismic denoising often lacks perfectly clean labels. This means that the quantitative metrics measure agreement with the provided targets, not necessarily geological correctness. Metrics such as MAE, MSE, PSNR, and SSIM are useful for comparison, but they do not fully capture whether weak reflectors, faults, or subtle stratigraphic patterns remain interpretable after denoising.

The router analysis is also limited. Gate maps show where the model prefers one expert over the other, but the link between gate values and specific geological structures is based on qualitative inspection rather than a dedicated quantitative gate-versus-structure study. The SSB is evaluated with two experts only; different backbones, more experts, or other router inputs could change both performance and gate behavior.

Finally, the method improves training efficiency by updating only a subset of parameters, making it cheaper to train than full fine-tuning or pretraining a foundation model from scratch. However, inference still requires evaluating multiple foundation-model ex-

perts and a router. This makes the joint SSB more expensive at inference than a single-expert model, even after inference optimization. The final evaluation also does not include external surveys, field-only validation, blind expert interpretation by geophysicists, or a full deployment study on large continuous seismic volumes.

6 Conclusion

6.1 Research Conclusions

This thesis presents a focused residual expert-aware SSB for seismic denoising. DINOv3 LoRA and SwinV2 LoRA are trained as standalone denoisers, then fused by a residual router. Router-only residual fusion improves over both standalone experts, and joint residual fine-tuning gives the best result with MAE 0.0463 ± 0.0046, MSE 0.0122 ± 0.0029, PSNR 19.14 ± 0.20 dB, and SSIM 0.9727 ± 0.0071.

The main conclusion is that the strongest final system is not a single adapted backbone, but a jointly trained residual spatial fusion model. Expert co-adaptation and residual expert-aware routing are key to the final performance.

6.2 Future Work

Future work should evaluate the method on additional seismic datasets and different noise types, including field data where ideal clean labels are not available. This would make it possible to study whether the observed gains transfer beyond the controlled supervised dataset used in this project. Evaluation on field data could also include blind comparison by domain experts, since standard metrics do not fully capture whether weak but geologically meaningful structures are preserved.

The router could also be extended with uncertainty estimates. Instead of only predicting a spatial gate between the two experts, the model could estimate where its denoised output is reliable and where the experts disagree in a meaningful way. This would be useful for seismic interpretation, because uncertain regions could be flagged for further inspection rather than treated as equally trustworthy predictions.

Finally, the trained SSB could be compressed through knowledge distillation. In this setup, the full two-expert SSB would act as a teacher model, while a smaller single-pass student denoiser would be trained to reproduce its outputs. This could reduce inference cost because deployment would no longer require evaluating both foundation-model experts and the router for every input patch. This compression would make the approach more practical where inference speed and memory usage are important, but it would require more compute for training.

7 Responsible Research

7.1 Reproducibility

All reported trained models are intended to be reproducible from the public code repository and the saved configuration files, checkpoints, and metric outputs generated by the training pipeline. The final implementation used for the reported experiments is stored in the public repository `AlexisFimeyer/research-project`. The main shared training code, model-specific configurations, and repeated-experiment manifest are:

- `src/modeling.py`
- `src/train.py`
- `configs/swin_lora.yaml`
- `configs/dinov3_unet_lora.yaml`
- `configs/router_only_ssb.yaml`
- `configs/joint_ssb.yaml`
- `configs/repeated_experiments.yaml`

The individual YAML files define the hyperparameters for one run, including the backbone choice, LoRA rank and scaling, optimizer settings, batch size, epoch count, loss weights, and data paths. The repeated-experiment manifest records the higher-level evaluation grid: five initialization seeds, 42–46, crossed with five dataset split seeds, 0–4. The helper scripts `src/prepare_splits.py` and `src/run_repeated.py` use this manifest to prepare the split directories and launch the corresponding training runs without manually editing configuration files.

The same split seeds are reused across SwinV2 LoRA, DINOv3 LoRA, and SSB variants so that model comparisons are paired by data partition and initialization seed. Reported means and standard deviations are computed from the saved evaluation files for these 25 repetitions. The frozen SFM-Base baseline is deterministic and is therefore reported as a point estimate. The evaluation scripts use saved checkpoints rather than selecting results manually from training logs.

Exact bitwise reproducibility is not guaranteed because training uses GPU kernels, mixed precision, and library implementations that can vary across CUDA, PyTorch, and hardware versions. To make this limitation explicit, this thesis reports performance over repeated seeds and splits instead of relying on a single run. Reproducing the results should therefore be understood as reproducing the same experimental protocol and comparable aggregate metrics, not necessarily identical floating-point values for every batch.

7.2 Computational Cost

The final repeated evaluation consisted of 25 complete seed/split combinations. Each combination included the reported trained model families: DINOv3 LoRA,

SwinV2 LoRA, residual router-only SSB, and residual joint SSB. Each model was trained for 50 epochs on a single NVIDIA GeForce RTX 4090 GPU with 24 GB memory, using mixed precision. Including repeated training, evaluation, and supporting runs, the final experimental campaign corresponds to approximately 60 GPU-hours.

Following standard footprint reporting practice [38], operational energy was estimated as $E = Pt$. Assuming the RTX 4090 operated at its maximum power of 450 W, this corresponds to 0.45 kW. For 60 h of GPU use, the GPU-only energy use is

$$0.45 \times 60 = 27.00\,\text{kWh}.$$

Adding an estimated 50 W, or 0.05 kW, for non-GPU system power gives a total system power estimate of

$$0.50 \times 60 = 30.00\,\text{kWh}.$$

Using the Netherlands 2026 grid-mix factor of 0.244 kgCO2e/kWh [39], this corresponds to

$$27.00 \times 0.244 = 6.59\,\text{kgCO2e},$$

for GPU-only energy, and

$$30.00 \times 0.244 = 7.32\,\text{kgCO2e},$$

for the system-level estimate. These values exclude emissions from hardware manufacturing.

7.3 Ethical Implications

This work does not use human-subject data, but seismic denoising can still affect real-world interpretation. An over-aggressive model may remove weak but meaningful reflectors, while remaining noise may be mistaken for geological structure. This matters in active-seismic applications such as geothermal exploration, groundwater assessment, and carbon-storage monitoring.

The main risk is poor robustness under distribution shifts. Real seismic data can differ across acquisition, geological regions, preprocessing pipelines, and noise types. Since the method adapts vision foundation models trained on natural images, the results should be validated on independent field data before practical use. LoRA and router-only blending reduce trainable parameters, but the use of large pretrained backbones still creates compute and accessibility costs.

7.4 Use of AI Tools

AI tools were used during the preparation of this thesis to support grammar, code debugging, LaTeX formatting, and figure/table editing, prompts can be found at Appendix section B. The tools were not used to generate experimental results. All reported metrics, model outputs, and conclusions are based on the implemented training and evaluation pipeline, saved checkpoints, and evaluation files. No scientific claims, experimental outcomes, or conclusions were taken directly from LLM-generated text.

Acknowledgements

I would like to thank my responsible professor, Dr. Jing Sun, and my supervisors, Dr. Tiexing Wang, Dr. Eric Verschuur, and Jiahua Zhao, for their guidance, feedback, and support throughout this research project.

I am grateful to Delft University of Technology and the EEMCS Faculty for providing the academic environment and resources that made this project possible.

References

[1] R. E. Sheriff and L. P. Geldart, *Exploration Seismology*, 2nd ed. Cambridge University Press, 1995. DOI: 10.1017/CBO9781139168359.

[2] G. Hennenfent and F. J. Herrmann, "Seismic denoising with nonuniformly sampled curvelets," *Computing in Science & Engineering*, vol. 8, no. 3, pp. 16–25, 2006. DOI: 10.1109/MCSE.2006.49.

[3] R. Neelamani, A. I. Baumstein, D. G. Gillard, M. T. Hadidi, and W. L. Soroka, "Coherent and random noise attenuation using the curvelet transform," *The Leading Edge*, vol. 27, no. 2, pp. 240–248, 2008. DOI: 10.1190/1.2840373.

[4] K. Chen and M. D. Sacchi, "Robust reduced-rank filtering for erratic seismic noise attenuation," *GEOPHYSICS*, vol. 80, no. 1, pp. V1–V11, 2015. DOI: 10.1190/geo2014-0116.1.

[5] L. Zhu, E. Liu, and J. H. McClellan, "Joint seismic data denoising and interpolation with double-sparsity dictionary learning," *Journal of Geophysics and Engineering*, vol. 14, no. 4, pp. 802–810, 2017. DOI: 10.1088/1742-2140/aa6491.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, ser. Lecture Notes in Computer Science, vol. 9351, Springer, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[9] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017. DOI: 10.1109/TIP.2017.2662206.

[10] H. Sheng, X. Wu, X. Si, J. Li, S. Zhang, and X. Duan, "Seismic foundation model: A next generation deep-learning model in geophysics," *GEOPHYSICS*, vol. 90, no. 2, pp. IM59–IM79, 2025. DOI: 10.1190/geo2024-0262.1.

[11] J. Zhao, U. bin Waheed, J. Sun, Y. Cui, N. Savva, and E. Verschuur, *Parameter-efficient adaptation of pre-trained vision foundation models for active and passive seismic data denoising*, 2026. DOI: 10.48550/arXiv.2605.10953. arXiv: 2605.10953 [physics.geo-ph].

[12] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008. [Online]. Available: https://papers.nips.cc/paper/7181-attention-is-all-you-need.

[13] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy.

[14] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9650–9660. DOI: 10.1109/ICCV48922.2021.00951.

[15] M. Oquab et al., *DINOv2: Learning robust visual features without supervision*, 2023. DOI: 10.48550/arXiv.2304.07193. arXiv: 2304.07193 [cs.CV].

[16] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009. DOI: 10.1109/CVPR52688.2022.01553.

[17] A. Kirillov et al., "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026. DOI: 10.1109/ICCV51070.2023.00371.

[18] O. Siméoni et al., *DINOv3*, 2025. DOI: 10.48550/arXiv.2508.10104. arXiv: 2508.10104 [cs.CV].

[19] Z. Liu et al., "Swin transformer v2: Scaling up capacity and resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 009–12 019.

[20] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97, 2019, pp. 2790–2799. [Online]. Available: https://proceedings.mlr.press/v97/houlsby19a.html.

[21] M. Jia et al., "Visual prompt tuning," in *European Conference on Computer Vision*, Springer, 2022, pp. 709–727. DOI: 10.1007/978-3-031-19827-4_41.

[22] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9.

[23] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991. DOI: 10.1162/neco.1991.3.1.79.

[24] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994. DOI: 10.1162/neco.1994.6.2.181.

[25] N. Shazeer et al., "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=B1ckMDqlg.

[26] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *Journal of Machine Learning Research*, vol. 23, no. 120, pp. 1–39, 2022. [Online]. Available: https://jmlr.org/papers/v23/21-0998.html.

[27] C. Riquelme et al., "Scaling vision with sparse mixture of experts," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 8583–8595. [Online]. Available: https://papers.nips.cc/paper/2021/hash/48237d9f2dea8c74c2a72126cf63d933-Abstract.html.

[28] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. DOI: 10.1109/TIP.2003.819861.

[29] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, vol. 2, 2003, pp. 1398–1402. DOI: 10.1109/ACSSC.2003.1292216.

[30] Y. Wu and K. He, "Group normalization," in *European Conference on Computer Vision*, 2018, pp. 3–19. DOI: 10.1007/978-3-030-01261-8_1.

[31] D. Hendrycks and K. Gimpel, *Gaussian error linear units (GELUs)*, 2016. DOI: 10.48550/arXiv.1606.08415. arXiv: 1606.08415 [cs.LG].

[32] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations*, 2016. [Online]. Available: https://arxiv.org/abs/1511.07122.

[33] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2881–2890. DOI: 10.1109/CVPR.2017.660.

[34] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, 1992. DOI: 10.1016/0167-2789(92)90242-F.

[35] NVIDIA Corporation. "GeForce RTX 4090 Graphics Cards for Gaming," Accessed: Jun. 1, 2026. [Online]. Available: https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/rtx-4090/.

[36] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[37] R. Wightman, *PyTorch Image Models*, 2019. [Online]. Available: https://github.com/huggingface/pytorch-image-models.

[38] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *Journal of Machine Learning Research*, vol. 21, no. 248, pp. 1–43, 2020. [Online]. Available: https://jmlr.org/papers/v21/20-312.html.

[39] CO2-emissiefactoren.nl, *Elektriciteit - stroom (onbekend) gridmix*, https://co2emissiefactoren.nl/factoren/2026/11/52/elektriciteit-stroom-onbekend-gridmix/?unit=kwh, Emission factor: 0.244 kg CO_2-eq/kWh. Accessed: 2026-06-01, 2026.

Appendix

A Bar Chart Results

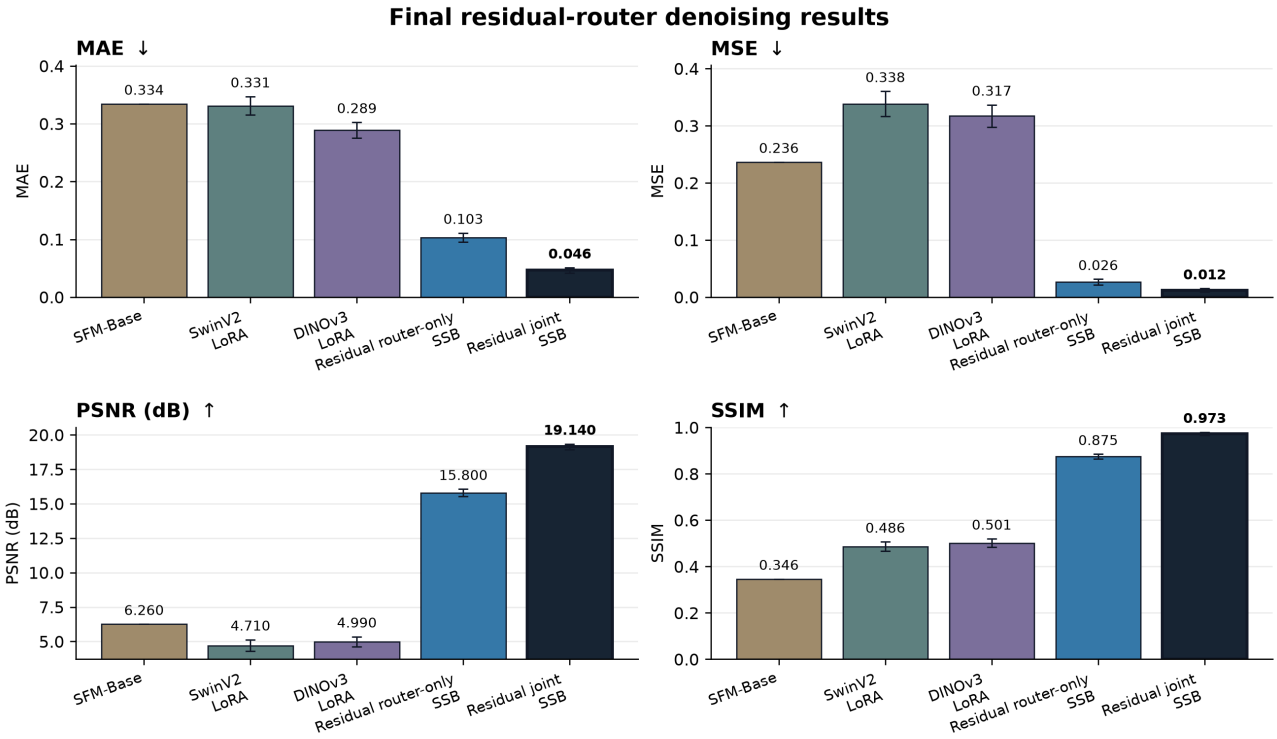

Figure 4: Metric comparison for the SFM-Base baseline, the two standalone experts, the residual router-only SSB, and the residual joint SSB. Error bars show the standard deviation over 25 seed/split repetitions for trained models; SFM-Base is deterministic. Models are ordered from worst to best by SSIM in every panel.

B Prompts Used for AI Tools

Example prompts that were used for AI tools are listed below. This is not an exhaustive list of prompts used during the project, but the examples indicate the type of assistance requested and the way large language models were used. The AI models consulted during the project included GPT-5.4, GPT-5.5, DeepSeek V4 Pro, DeepSeek V4 Flash, MiniMax M3, Claude Opus 4.7, Claude Opus 4.8 and Claude Fable 5.

B.1 Writing Support

- Rewrite this paragraph in a more formal academic writing style. Do not change the meaning or add new claims.
- Make this explanation of the soft spatial blender clearer and more concise.
- Check this section for grammar and sentence flow, while keeping my original structure.
- Help me formulate the limitations section without overstating the results.

B.2 Programming Support

- Can you help me find the error in this Python script that loads evaluation JSON files and extracts MAE, MSE, PSNR, and SSIM into a clean readable table.
- Can you verify that this function exists in this version of PyTorch.

B.3 Literature Support

- Help me find literature about parameter-efficient fine-tuning methods such as LoRA and adapters.
- Help me find papers about vision foundation models and their transfer to dense prediction tasks.

B.4 Visualization Support

- Please suggest a clear figure layout for showing the noisy input, clean target, model prediction, residual, and router gate.
- Please help me make the captions for my architecture and router figures more precise.

B.5 Formatting Support

- How do I make appendix sections appear as A, B, C, and D in an article document?
- How can I resize the table to not be in a two column format?