FROM PARALLELISM TO SPECIALISATION: SSA-GUIDED MULTI-BRANCH IMPLICIT NEURAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Although Multi-branch Implicit Neural Representations (INRs) increase the modelling capacity of the network, the parallel paths are not inherently guided to learn distinct components of the target signal. To address this limitation, we introduce a Singular Spectrum Analysis (SSA) guided multi-branch INR, which provides each branch with a specific signal component as the branch-specific target. For a two-dimensional signal, for example, an image, a separable two-dimensional SSA decomposes the target signal into 4 distinct components, which separately represent structure, horizontal-detail, vertical-detail, and joint-detail, to provide each branch with a designated target to achieve. Gated contributions of each branch are then additively combined to obtain the final signal reconstruction. In addition, we introduce a Particle Swarm Optimisation (PSO)-initialised, learnable coordinate scaling at the input to each branch to enable it to model the target effectively. Branch-wise pretraining against the respective SSA component ensures that all the parallel branches are properly supervised to attain the respective target. Experiments on Kodak images demonstrate that the proposed algorithm shows the best overall peak signal-to-noise ratio (PSNR) performance among state-of-the-art (SOTA) algorithms including COSMO-RC, INCODE, FR-INR, and SIREN. Moreover, ablation studies against unguided parallel INRs show that the guided model consistently outperforms the unguided baseline. Spatial alignment matrices, training dynamics, and Fourier-domain analysis suggest that the learned branches retain distinct and coherent roles. The same specialisation principle extends to any n-dimensional signal, including audio, which is a one-dimensional signal. Further ablations demonstrate that the main source of the improvement is branch specialisation, indicating that specialisation and explicit branch-wise task allocation are more important than branch parallelisation alone. The complete source code and implementation will be made publicly available upon acceptance.

1 INTRODUCTION

Implicit Neural Representations (INRs) model discrete signals as continuous implicit functions, parametrised by neural networks (NNs) (Sitzmann et al., 2020), which map input coordinates to signal values, such as image coordinates to RGB intensities, temporal coordinates to audio amplitudes, or spatial-temporal coordinates to physical field measurements (Sitzmann et al., 2020; Mildenhall et al., 2020). INRs are used to model various types of signals with different dimensions, such as audio signals, images, neural fields, and physical signals. They enable storing the continuous representations of a signal independent of the signal's resolution. Despite these advantages, INRs often suffer from spectral bias: the tendency of NNs to learn low-frequency information more readily than rapidly varying high-frequency information (Rahaman et al., 2019; Tancik et al., 2020). In an image, for example, NNs tend to learn smooth, flat regions of the image more readily than sharp edges, fine textures, and local geometric details.

Transforming input coordinates before passing them to the network is a viable approach to mitigate spectral bias. By the Fourier scaling property, scaling the input coordinates in the spatial or the time domain results in an inverse scaling in the frequency domain (Oppenheim et al., 1999), which enables controlling the accessible frequency features to the network Tancik et al. (2020). However,

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.

a single scaling is unlikely to suit all the varied structural components of a signal. For instance, smooth color variations, directional edges, and textures of an image contain distinct spectral characteristics. This motivates using multi-branch INR architectures, which work in parallel to model different spectral components. Accordingly, each branch views the signal through different coordinate transformations (Li et al., 2024).

Our main argument is that parallelisation alone does not imply specialisation. When a multi-branch architecture learns solely using a reconstruction objective, it only learns to reconstruct the target signal. Individual branches are not supervised to perform a specific task during the process. This can cause the branches to learn redundant content and exchange tasks during the optimisation. In addition, this lack of explicit supervision allows the fusion mechanism to compensate for weak branch outputs. Although the network contains parallel expert subnetworks architecturally, their functional responsibilities are not explicitly specialised (Wang et al., 2022; Ben-Shabat et al., 2024). Thus, we argue that multi-branch networks require explicit task allocation.

In this study, we are providing the branches with such explicitly allocated responsibilities using two-dimensional Singular Spectrum Analysis (SSA) as a data-adaptive signal decomposition tool (Broomhead & King, 1986; Golyandina & Zhigljavsky, 2013). The underlying principle of our study lies in classical decompositions such as Laplacian Pyramids and Wavelets, which separate signals into components with different spatial and spectral characteristics (Burt & Adelson, 1983; Mallat, 1989). Building on that principle, we use SSA to decompose the target signal along spatial and spectral scales. For instance, an image is decomposed along two spatial axes and into two spectral scales along each axis, resulting in a $2^2 = 4$ component decomposition.

We use two-dimensional SSA to assign branch roles. Decomposing an image along spatial axes yields structural, x-detail, y-detail, and joint detail targets for branches S, X, Y, and D. Each branch produces an RGB approximation and a gate logit. The branch contribution is supervised by the corresponding target, and the contributions are added to obtain the final reconstruction. SSA-derived activity maps guide where branches contribute. PSO-initialised learnable coordinate scales adapt each branch's accessible frequencies to the signal.

Coordinate scaling determines the range of spectral content accessible to a given branch. However, the appropriate scale depends on the signal itself, as different signals contain distinct spectral characteristics. This motivates using learnable coordinate scaling, which allows each branch to adapt its bandwidth according to the data. A Particle Swarm Optimisation (PSO)-based initialisation is used to provide a data-dependent initialisation for coordinate scales, which is then refined using gradient-based optimisation algorithms (Kennedy & Eberhart, 1995). In summary, SSA defines the responsibility of each branch, while learnable coordinate scaling controls how a branch accesses its assigned frequency components.

The main contributions of this work are as follows:

1. Moving beyond architectural parallelism, we introduce branch-specific targets and an independently gated additive fusion mechanism to control the contribution of each branch at a given spatial location.
2. To generate these targets, we formulate a separable two-dimensional SSA decomposition, which decomposes a given signal along its coordinate axes into low and high variation components, providing direct branch-level supervision to train the branches individually.
3. We incorporate learnable coordinate scaling at each branch input to align the network's accessible bandwidth with different spectral characteristics of the target signal. Coordinate scales are initialised using PSO and subsequently refined using gradient-based optimisation.
4. Finally, we evaluate the proposed method through experiments in image and audio representation, including branch alignment, frequency-domain analysis, training dynamics, and ablation studies.

2 Related Work

INRs represent discrete signals as continuous functions, which map signal coordinates to signal values. Sitzmann et al. (2020) introduce the efficiency of using periodic activations in modelling natural

signals and their derivatives. NNs' inherent property of learning low frequencies more readily than high frequencies is explained by Spectral Bias (Rahaman et al., 2019). According to Tancik et al. (2020), Fourier features use periodic coordinate embedding to mitigate this issue. However, it does not allocate signal components across specialised modules.

2.1 Frequency-Aware and Multi-Scale INRs

Frequency-aware INRs use modified activation functions, coordinate encodings, and training parametrisation to improve the ability of representing high frequency components. Saragadam et al. (2023) use complex Gabor wavelets, while Liu et al. (2023) propose variable periodic activations. Kazerouni et al. (2024) use prior knowledge embeddings to condition sinusoidal activations, whereas Shi et al. (2024) mitigate spectral bias using Fourier reparametrised training. Thennakoon et al. (2026) introduce complex sinusoidal modulation with their regularised raised cosine architecture. Related approaches include Adaptive Wavelet-Positional Encoding Zhao et al. (2025), SCONE Li et al. (2024), Multi-Scale Sine Activation Han et al. (2026), and I-INR Haider et al. (2026). Although these approaches improve spectral access, they typically supervise only the final reconstruction and do not explicitly define component-level roles.

2.2 Multi-Branch and Modular Coordinate Networks

KiloNeRF partitions radiance fields across many small MLPs (Reiser et al., 2021). Neural Implicit Dictionary Learning proposed by Wang et al. (2022) uses learned coordinate-network bases, while MoE-based INRs combine local experts through learned routing (Ben-Shabat et al., 2024; Zhao et al., 2023). In these methods, specialised roles are often determined implicitly by spatial partitioning, routing, or reconstruction loss. Instead, our method uses SSA to supervise each branch explicitly using data-dependent targets.

2.3 Signal-Decomposition-Guided Representation Learning

Signal decompositions organise image content in different ways. Cooley & Tukey (1965) represent global spatial frequencies by Fourier transform, and it can be efficiently computed using the fast Fourier transform (FFT). The two-dimensional discrete wavelet transform (DWT) provides spatially localised approximation and directional-detail subbands (Mallat, 1989). The Karhunen–Loève transform (KLT) derives a basis ordered by the variance from the signal covariance (Karhunen, 1947). In contrast, SSA extracts data-adaptive components from low-rank trajectory matrices (Broomhead & King, 1986; Golyandina & Zhigljavsky, 2013). INR methods use multiscale structure through networks inspired by Laplacian Pyramids as in MINER (Saragadam et al., 2022), or band-limited coordinate networks, as in BACON (Lindell et al., 2022). However, in this work, we use four decompositions of the target signal to supervise branches directly. For that, we use separable two-dimensional SSA and we compare it with two-dimensional FFT, two-dimensional DWT, and separable KLT as alternative signal decomposition methods.

3 Methods

Scope of the formulation. We present the method for images, in which two spatial directions and two frequency divisions yield four parallel branches in its simplest form: branch S representing structural content, branch X representing x-coordinate detail, branch Y representing y-coordinate detail, and branch D representing joint detail. More generally, it is possible to extend the same methodology to an N-dimensional signal by assigning branch-wise responsibilities along the signal axes and frequency ranges. The N-dimensional formulation is discussed for audio and higher-dimensional signals, and the details are presented in Appendix A.4.

3.1 Problem Formulation

Let $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ be an RGB image and $\boldsymbol{x} = (x, y) \in [-1, 1]^2$ a spatial coordinate. Four branches $\mathcal{B} = \{S, X, Y, D\}$ are assigned to model structure, x-detail, y-detail, and joint detail. Each branch is supervised by the corresponding SSA target rather than relying solely on the final reconstruction loss. The generalisation for N-dimensional signals is given in appendix A.4.

3.2 SEPARABLE SSA TARGET RECONSTRUCTION

To assign explicit targets to the branches, we decompose the target image using SSA. A separable two-dimensional SSA is used by applying one-dimensional SSA along the spatial axes and RGB channels. Details of the one-dimensional SSA operation, including the trajectory matrix, low-rank reconstruction, and diagonal averaging, are provided in the Appendix A.1.

First, SSA is applied along the horizontal direction to split the image into low- and high-frequency contents along the x-direction. It gives $I \rightarrow I_{xL} + I_{xh}$. Subsequently, another one-dimensional SSA is applied along the vertical direction to both of the intermediate SSA components (I_{xL} and I_{xH}) separately. This gives: $I_{xL} \rightarrow I_{LL} + I_{LH};\ I_{xH} \rightarrow I_{HL} + I_{HH}$

For branch supervision, derived SSA components are relabelled by their functional roles.

$$I_S^{SSA} = I_{LL}; \quad I_X^{SSA} = I_{HL}; \quad I_Y^{SSA} = I_{LH}; \quad I_D^{SSA} = I_{HH}$$

Here, S denotes smooth structural components, X denotes the detail primarily associated with variations along the x-axis, Y denotes the detail primarily associated with variations along the y-axis, and D denotes the joint high-variation detail.

Therefore, the branch target set is given by:

$$\{I_b^{SSA}\}_{b\in\mathcal{B}} = \{I_S^{SSA}, I_X^{SSA}, I_Y^{SSA}, I_D^{SSA}\}$$

With that, we can convert the branch specialisation problem into a set of supervised component learning problems.

The one-dimensional SSA derivation and a visual summary of the separable target construction are presented in Appendix A.2.

It is important to note that one may use any theoretically appropriate signal decomposition method to obtain the branch targets and gate priors. A comparative analysis was performed using two-dimensional FFT, two-dimensional DWT, and KLT, and further details are presented in Appendix A.8. However, the central contribution lies in specialisation; SSA was used to construct branch targets and gate priors.

3.3 COORDINATE-SCALED BRANCH ARCHITECTURE

Each role-specific branch $b \in \mathcal{B}$ is implemented as a SIREN-style network $F_b : \mathbb{R}^2 \rightarrow \mathbb{R}^4$.

Instead of receiving the original coordinate vector $\boldsymbol{x}$, each branch receives a scaled version of $\boldsymbol{x}$, which is obtained by $\tilde{\boldsymbol{x}}_{\boldsymbol{b}} = \mathbf{s}_b \odot \boldsymbol{x}$.

where $\mathbf{s}_b = (a_b, b_b)$ is the branch-specific coordinate scale vector, and $\odot$ is the element-wise multiplication operator.

Each branch $b \in \mathcal{B}$ predicts an RGB component ($\mathbf{y}_b(\boldsymbol{x}) \in \mathbb{R}^3$), which is an approximation of the assigned target, and a scalar gate logit ($\alpha_b(\boldsymbol{x}) \in \mathbb{R}$).

The default scale initialisations are aligned with their tasks as follows $\mathbf{s}_S = (1, 1);\ \mathbf{s}_X = (m, 1);\ \mathbf{s}_Y = (1, m);\ \mathbf{s}_D = (m, m)$ where m is the high-frequency multiplier.

The coordinate scale vectors are learnable and parametrised in log space:

$$\mathbf{s}_b = \exp(\eta_b) \tag{1}$$

thereby keeping the effective coordinate scales positive during the training process.

3.4 SSA-GUIDED GATED ADDITIVE FUSION

The resulting gate logits are then converted into a set of independent sigmoid gates.

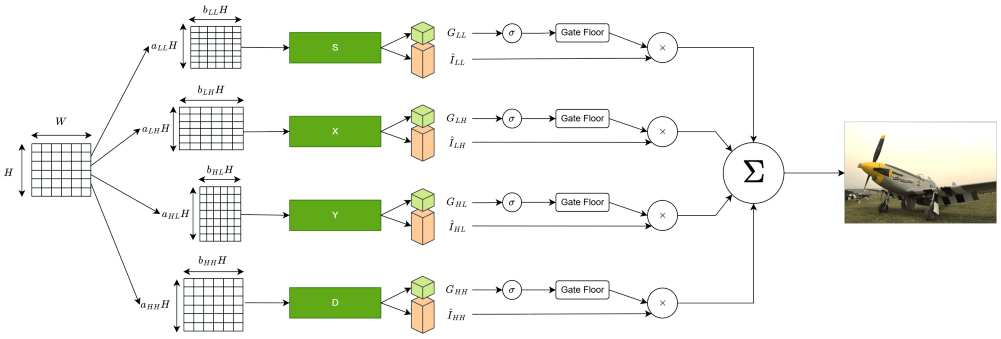


Figure 1: Overview of the proposed SSA-guided multi-branch INR. A branch-specific scaled coordinate input is given to each role-defined branch $b \in \{S, X, Y, D\}$ and predicts an RGB component together with an independent gate. The gated branch contribution $c_b = g_b y_b$ is supervised by the corresponding SSA-derived target I_b^{SSA}. The SSA-derived gate prior G_b^{SSA} guides the learned gate g_b The final image is reconstructed by adding all gated branch contributions.

$$g_b(\boldsymbol{x}) = g_{min} + (1 - g_{min}) \left(\frac{\alpha_b(\boldsymbol{x})}{\tau} \right) \tag{2}$$

where τ is the gate temperature, and g_{min} is a small gate floor to ensure that every branch contributes to the final output.

Accordingly, the effective branch contributions are defined as $\mathbf{c}_b(\boldsymbol{x}) = g_b(\boldsymbol{x})\mathbf{y}_b(\boldsymbol{x})$.

The final reconstruction is the sum of the branch contributions.

$$\hat{\mathbf{I}}(\boldsymbol{x}) = \sum_{b \in \mathcal{B}} \mathbf{c}_b(\boldsymbol{x}) \tag{3}$$

Sigmoid gates are used instead of softmax gates, which impose the sum-to-one condition, since SSA components are inherently additive and different spectral features may coexist at the same spatial location.

3.5 SSA-Derived Gate Priors

In addition to the branch targets, SSA provides soft spatial priors indicating how strongly each branch should contribute at each spatial location. We compute a normalised activity map from the corresponding SSA target ($\mathbf{I}_b^{SSA}$) for each branch $b \in \mathcal{B}$.

$$G_b^{SSA}(\mathbf{x}) = \left(\frac{\frac{1}{3}\sum_{c=1}^{3} |I_b^{SSA}(\mathbf{x}, c)|}{\max_{j \in \mathcal{B}} \left(\frac{1}{3}\sum_{c=1}^{3} |I_j^{SSA}(\mathbf{x}, c)| \right) + \epsilon} \right)^{\gamma} \tag{4}$$

where $c \in \{1, 2, 3\}$ is the RGB channel index, $\epsilon > 0$ is a numerical-stability constant, and $\gamma > 0$ is an exponent controlling the contrast of the prior maps.

The same gate floor defined in equation equation 2 is applied here.

The complete architecture is summarised in Figure

$$\tilde{G}_b^{SSA} = g_{min} + (1 - g_{min}) G_b^{SSA} \tag{5}$$

The resulting gate prior vector is defined as:

$$\tilde{G}^{SSA}(\mathbf{x}) = [\tilde{G}_S^{SSA}(\mathbf{x}), \tilde{G}_X^{SSA}(\mathbf{x}), \tilde{G}_Y^{SSA}(\mathbf{x}), \tilde{G}_D^{SSA}(\mathbf{x})] \in [0,1]^4$$

The derivation and implementation details of constructing gate priors are provided in Appendix A.3.

3.6 Training Objective

The training objective combines reconstruction and SSA targets, gate prior guidance, and a weak amplitude regularisation. The reconstruction loss is given by:

$$\mathcal{L}_{rec} = \frac{1}{|\Omega_d|} \sum_{\boldsymbol{x} \in \Omega_d} \left\| \hat{\mathbf{I}}(\boldsymbol{x}) - \mathbf{I}(\boldsymbol{x}) \right\|_2^2 \tag{6}$$

Each branch is trained to approximate the corresponding SSA target by supervising the gated branch contribution rather than the raw branch output.

$$\mathcal{L}_{SSA} = \sum_{b \in \mathcal{B}} \frac{1}{|\Omega_d|} \sum_{\boldsymbol{x} \in \Omega_d} \left\| \mathbf{c}_b(\boldsymbol{x}) - I_b^{SSA}(\boldsymbol{x}) \right\|_2^2 \tag{7}$$

The learned gate values are supervised by the SSA-derived gate priors obtained from equation equation 4.

$$\mathcal{L}_{gate} = \frac{1}{|\Omega_d|} \sum_{\boldsymbol{x} \in \Omega_d} \left\| \mathbf{G}^{SSA}(\boldsymbol{x}) - \mathbf{G}(\boldsymbol{x}) \right\|_2^2 \tag{8}$$

where

$$\mathbf{G}(\boldsymbol{x}) = [g_S(\boldsymbol{x}), g_X(\boldsymbol{x}), g_Y(\boldsymbol{x}), g_D(\boldsymbol{x})]$$

Under this formulation, the optimisation algorithm has the flexibility to tune branch contributions by adjusting the RGB outputs and gate values through compensating adjustments, introducing a scale ambiguity. To reduce this ambiguity, we use a weak raw-amplitude penalty.

$$\mathcal{L}_{amp} = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \mathbb{E}_{\boldsymbol{x} \in \Omega_d} [\|\mathbf{y}_b(\boldsymbol{x})\|_2^2] \tag{9}$$

The full objective is:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{SSA}(t)\mathcal{L}_{SSA} + \lambda_g(t)\mathcal{L}_{gate} + \lambda_a \mathcal{L}_{amp} \tag{10}$$

where t denotes the epoch.

$\lambda_{SSA}(t)$ and $\lambda_g(t)$ decay during training to make SSA guidance strong in the early stages of training. Later, optimisation refines the final reconstruction. In the implementation, the weighted SSA term is capped relative to $\mathcal{L}_{rec}$ for stability.

3.7 Learnable Coordinate Scaling and PSO Initialisation

The optimal scale varies with the signal, so its log parameters are learned during the training. Before branch pretraining stage, PSO is used to search eight scale coordinates $\boldsymbol{q} = [a_S, b_S, a_X, b_X, a_Y, b_Y, a_D, b_D]$. Scores are given to each candidate using a short fitting stage guided by SSA. The best candidate is selected as the optimal initialisation for gradient-based optimisation.

Table 1: Branch–target alignment for the proposed SSA-guided model. The clear diagonal structure indicates that each branch aligns most strongly with its assigned target.

	$\mathbf{I}_{\mathrm{S}}^{\mathrm{SSA}}$	$\mathbf{I}_{\mathrm{X}}^{\mathrm{SSA}}$	$\mathbf{I}_{\mathrm{Y}}^{\mathrm{SSA}}$	$\mathbf{I}_{\mathrm{D}}^{\mathrm{SSA}}$
$\mathbf{c}_{\mathrm{S}}$	**9.156e+01**	6.683e+05	6.706e+05	6.684e+05
$\mathbf{c}_{\mathrm{X}}$	6.685e+05	**1.584e+02**	2.157e+03	5.227e+02
$\mathbf{c}_{\mathrm{Y}}$	6.707e+05	2.027e+03	**1.592e+02**	2.068e+03
$\mathbf{c}_{\mathrm{D}}$	6.687e+05	5.068e+02	1.981e+03	**2.100e+02**

3.8 Branch-Specialised Training and Joint Optimisation

After scale initialisation, training proceeds in two stages. During the first stage, each branch is pretrained against its SSA target separately.

$$\mathcal{L}_{spec,b} = \left\|\mathbf{c}_b - I_b^{SSA}\right\|_2^2 + \lambda_{pg} \left\|g_b - \tilde{G}_b^{SSA}\right\|_2^2 \tag{11}$$

Here, λ_{pg} weights gate-prior supervision during branch pretraining. After that, all branches are jointly optimised with the full objective. Its decaying SSA and gate-prior terms retain early role guidance, while reconstruction loss drives the final output.

4 Experiments and Analysis

4.1 Experimental setup

We evaluate image reconstruction on all 24 Kodak (Eastman Kodak Company) images using full-image PSNR, comparing with COSMO-RC, INCODE, FR-INR, SIREN. Furthermore, we compare the proposed method with an architecturally matched unguided four-branch INR. We assess branch roles through target alignment, training dynamics, and Fourier spectra. The one-dimensional audio experiment is described in Appendix A.6.

4.2 Implementation details

Each of the four branches has three sinusoidal hidden layers of width 256. We use 500 epochs of branch-wise pretraining followed by 500 epochs of joint optimisation. Full implementation and evaluation details are given in appendix A.5.1.

4.3 Evidence of branch specialization

One of the core questions is, does the SSA guidance assign distinct responsibilities to each branch rather than just improving the fused output? The branch target alignment is quantified using $M_{ij} = \|c_i - I_j^{\mathrm{SSA}}\|_2^2$, where rows are learned gate contributions while the columns are SSA-targets. A lower value indicates a stronger alignment. The minimum value of each row is highlighted in bold. Figure 3 shows that the learned gate contributions c_S, c_X, c_Y, c_D retain structure, details in the x direction, the y direction, and jointly, as assigned by SSA. Since the final additive reconstruction is done by the gated components, the correspondence indicates the actual branch contribution, and the data in Table 1 and Table 2 support this observation. The lowest alignment error can be observed for each branch when it is measured against its assigned SSA targets, showing a clear diagonal dominance. Architecturally matched unguided model show a more ambiguous agreement between branches and targets and are less diagonal in nature. This shows higher redundancy than the explicit task allocation. The benefit of specialization is isolated from any benefit of parallelization by the control comparison.

Figure 3(a) analyzes whether this organization remains during joint optimization. While training, the PSNR value increases, while the SSA loss stays bounded and decreases after its initial adjustment. Complementary frequency-domain evidence is given in Figure 3(b).The S]contribution is focused around the low frequencies, the X and Y contributions show directional anisotropy, while the D

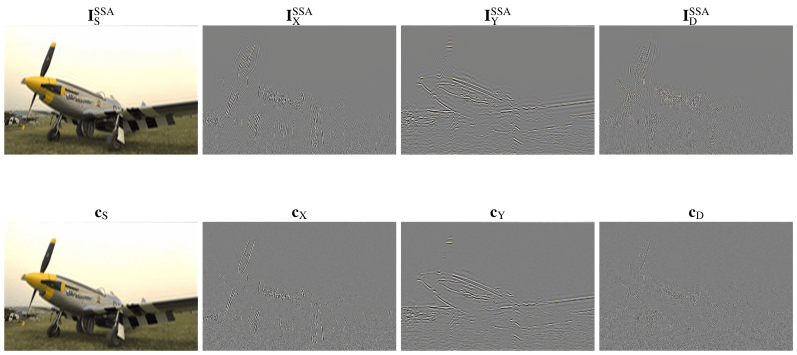
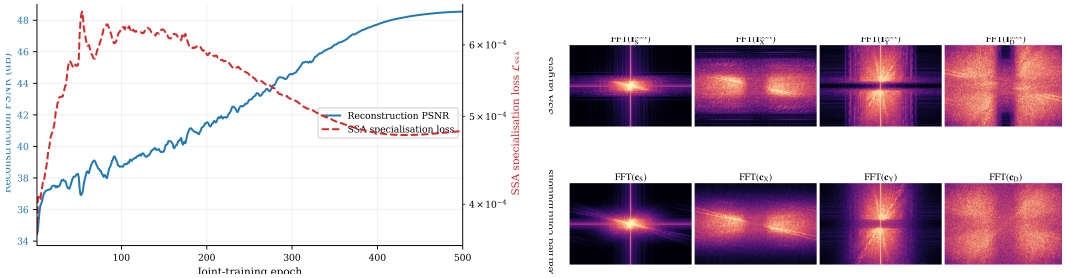


Figure 2: Evidence of branch specialization. The first row shows the SSA-derived component targets I_S^{SSA}, I_X^{SSA}, I_Y^{SSA} and I_D^{SSA} The second row shows the corresponding learned gated branch contributions c_S, c_X, c_Y and c_D. the close resemblance indicates that each branch has learned the role assigned by SSA-guided supervision.



(a) Joint-training dynamics

(b) Frequency-domain branch specialisation

Figure 3: Evidence of branch specialization. (a)While training the PSNR value increase and the SSA loss stays bounded and decreases after its initial adjustment.(b) Fourier magnitude spectra of the SSA targets and learned gated branch contributions. The learned branches recover the intended spectral roles of the structural, horizontal-detail, vertical-detail, and joint-detail components.

contribution has a wider spectral support associated with joint detail. The learned spectrum closely resembles the SSA-target spectrum in each case, confirming that the spatially assigned roles are consistent spectrally as well. The spatial, quantitative, temporal, and spectral evidence together shows genuine branch specialization rather than a decomposition of the fused output.

4.4 Representation performance and generalization

The proposed method achieves the highest PSNR among SOTA algorithms across the 24 Kodak images, and it outperforms the unguided parallel baseline on every image. Since the parallalisation is introduced from the unguided network, this result supports the paper's central premise on specialisation beyond parallel capacity.

Audio Representation. We also tested the two-branch formulation on a one-dimensional audio signal; its waveform, spectra, and branch contributions are reported in A.6.

4.5 Ablation Study

A cumulative ablation study is presented in table 2

The largest initial improvement can be seen when SSA branch supervision is added to the baseline unguided parallel model. This shows that branch-wise responsibility assignment provides substantial improvement. Branch pretraining introduces another significant improvement in PSNR, confirming the value of early division of responsibilities before joint optimisation. PSO scale initialisation,

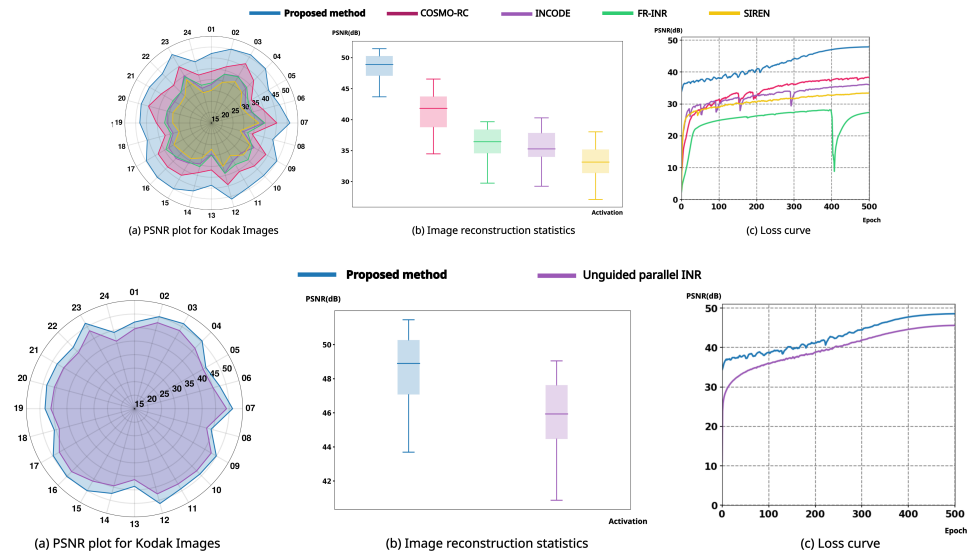


Figure 4: Image representation results on the 24 Kodak images. The top row compares the proposed method with COSMO-RC, INCODE, FR-INR, and SIREN, while the bottom row compares it with the architecturally matched unguided parallel INR.

Table 2: Cumulative ablation on image representation. Each row adds one component to the unguided multi-branch baseline. Higher PSNR is better.

Configuration	**PSNR (dB)** ↑
Unguided multi-branch INR	45.50
+ SSA branch loss	47.01
+ Branch-specialised training	48.16
+ PSO initialisation	48.37
+ SSA gate priors	48.45
+ Learnable coordinate scalars, full model	**48.61**

gate-priors, and learnable coordinate scaling show smaller but consistent improvements to the final reconstruction quality.

Here, one should understand that although adding parallel branches improved the PSNR when compared to classical single-branch models, explicit supervision shows a significant further improvement, supporting the central premise presented throughout the paper.

CONCLUSION

In our work, we proposed a multi-branch parallel INR network in which each branch in the network is explicitly guided to model distinct signal components through SSA-derived targets and gate priors.

Experiments performed on image and audio signals comparing the unguided parallel baseline and the proposed network show that the introduction of specialised branch-specific targets and gate priors improved the model performance by a significant margin, while learnable coordinate scaling and PSO scale initialisation further refined the final reconstruction. Furthermore, SSA decomposition of the target signal made the branch contributions more interpretable.

Future work can be extended to use the proposed algorithm in inverse tasks such as inpainting and image denoising. This requires SSA guidance to be consistent with the observations for incomplete signals and objectives that separate clean structure from random corruption.

AI USE STATEMENT

The research idea and the initial methodology were originally developed by the authors. Subsequently, OpenAI's ChatGPT was used to brainstorm and reaffirm the novelty of the idea through literature surveys. ChatGPT was used to assist with coding, implementation, and debugging.

The manuscript text was written by the authors. AI models were not used to draft or rewrite it. The Grammarly Chrome extension was used for grammar corrections, and the authors reviewed each suggestion before accepting.

We reviewed AI-assisted work, including methods, results, implementation, and debugging, and the authors are responsible for the final content.

REPRODUCIBILITY STATEMENT

Sections 3.2–3.8 specify the model, SSA-guided objectives, PSO scale initialisation, and training stages. Appendices A.1–A.3 detail the SSA decomposition and gate-prior construction, while Section 4.2 and Appendix A.5 report the image preprocessing, hyperparameters, evaluation protocol, and per-image Kodak results. The supplied train_SSA_PSO.py implements the Kodak image pipeline and saves checkpoints, evaluation metrics, and a summary CSV. Given the 24 Kodak images, the principal image experiment can be rerun with run_all_kodak –use_pso_scale_init; the PSO flag is required because it is disabled by default. Checkpoints are selected using full-image reconstruction error. Exact numerical agreement may vary because the script does not fix random seeds or package versions.

The model, SSA-guided targets, and PSO initialisation are specified in the sections 3.2–3.8. Details of the SSA decomposition and gate-prior construction are presented in Appendices A.1–A.3, while the image preprocessing, hyperparameters, evaluation protocols, and Kodak results are reported in section A.5.1 and appendix A.5. Appendix A.5.4 describes the external baseline comparison protocol. The implementation code for the proposed method will be made publicly available upon acceptance.

REFERENCES

Yizhak Ben-Shabat, Chamin Hewa Koneputugodage, Sameera Ramasinghe, and Stephen Gould. Neural experts: Mixture of experts for implicit neural representations. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

David S. Broomhead and Gregory P. King. Extracting qualitative dynamics from experimental data. *Physica D: Nonlinear Phenomena*, 20(2–3):217–236, 1986. doi: 10.1016/0167-2789(86)90031-X.

Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. doi: 10.1109/TCOM.1983.1095851.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. doi: 10.1090/S0025-5718-1965-0178586-1.

Eastman Kodak Company. Kodak lossless true color image suite. Kodak Lossless True Color Image Suite. Accessed: 2026.

Nina Golyandina and Anatoly Zhigljavsky. *Singular Spectrum Analysis for Time Series*. Springer, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-34913-3.

Ali Haider, Muhammad Salman Ali, Maryam Qamar, Tahir Khalil, Soo Ye Kim, Jihyong Oh, Enzo Tartaglione, and Sung-Ho Bae. I-inr: Iterative implicit neural representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4520–4528, 2026. doi: 10.1609/aaai.v40i6.42451.

Jufeng Han, Shu Wei, Min Wu, Lina Yu, Weijun Li, Linjun Sun, Hong Qin, and Yan Pang. Implicit neural representation with multi-scale sine activation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21567–21575, 2026. doi: 10.1609/aaai.v40i26.39305.

Kari Karhunen. Über lineare methoden in der wahrscheinlichkeitsrechnung. *Annales Academiae Scientiarum Fennicae, Series A I: Mathematica-Physica*, (37):1–79, 1947.

Amirhossein Kazerouni, Reza Azad, Alireza Hosseini, Dorit Merhof, and Ulas Bagci. INCODE: Implicit neural conditioning with prior knowledge embeddings. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1298–1307, 2024.

James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 – International Conference on Neural Networks*, volume 4, pp. 1942–1948. IEEE, 1995. doi: 10.1109/ICNN.1995.488968.

Jason Chun Lok Li, Chang Liu, Binxiao Huang, and Ngai Wong. Learning spatially collaged fourier bases for implicit neural representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13492–13499, 2024. doi: 10.1609/aaai.v38i12.29252.

David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16252–16262, 2022.

Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. Finer: Flexible spectral-bias tuning in implicit neural representation by variable-periodic activation functions, 2023.

Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. doi: 10.1109/34.192463.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision*, pp. 405–421. Springer, 2020.

Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, 2 edition, 1999.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 2019.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14335–14345, 2021.

Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G. Baraniuk, and Ashok Veeraraghavan. Miner: Multiscale implicit neural representation. In *Proceedings of the European Conference on Computer Vision*, pp. 318–333. Springer, 2022.

Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G. Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18507–18516, 2023.

Kexuan Shi, Xingyu Zhou, and Shuhang Gu. Improved implicit neural representation with fourier reparameterized training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25985–25994, 2024.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473, 2020.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7537–7547, 2020.

Pandula Thennakoon, Avishka Ranasinghe, Mario De Silva, Buwaneka Epakanda, Roshan Godaliyadda, Mervyn Ekanayake, and Vijitha Herath. Cosmo-inr: Complex sinusoidal modulation for implicit neural representations. In *Proceedings of the International Conference on Learning Representations*, 2026.

Peihao Wang, Zhiwen Fan, Tianlong Chen, and Zhangyang Wang. Neural implicit dictionary learning via mixture-of-expert training. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22613–22624. PMLR, 2022.

Hongxu Zhao, Zelin Gao, Yue Wang, Rong Xiong, and Yu Zhang. Adaptive wavelet-positional encoding for high-frequency information learning in implicit neural representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10430–10438, 2025. doi: 10.1609/aaai.v39i10.33132.

Jianchen Zhao, Cheng-Ching Tseng, Ming Lu, Ruichuan An, Xiaobao Wei, He Sun, and Shanghang Zhang. Moec: Mixture of experts implicit neural compression, 2023.

A Appendix

A.1 One-dimensional Singular Spectrum Analysis

In this section, we discuss the one-dimensional Singular Spectrum Analysis (SSA) used to obtain branch-specific targets. For a given one-dimensional signal, SSA embeds overlapping signal segments into a trajectory matrix and applies a low-rank decomposition. Then, we reconstruct a selected set of eigentriples before performing diagonal averaging to map the reconstructed matrix back to the signal domain. The retained reconstruction is taken as the low-variation component, whereas the residual is taken as the high-variation component.

A.1.1 Signal and Embedding

Let

$$\boldsymbol{z} = [z_1, z_2, \ldots, z_T]^\mathsf{T} \in \mathbb{R}^T \tag{12}$$

be a real-valued one-dimensional signal of length T. A window length L is selected in such a way that

$$2 \leq L \leq T - 1$$

Then define,

$$K = T - L + 1 \tag{13}$$

the lagged vectors are:

$$\boldsymbol{x}_k = [z_k, z_{k-1}, \ldots, z_{k+L-1}]^\mathsf{T} \in \mathbb{R}^L, \qquad k = 1, \ldots, K$$

A trajectory matrix is formed with these vectors.

$$\boldsymbol{X} = [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \cdots \quad \boldsymbol{x}_K] \in \mathbb{R}^{L \times K} \tag{14}$$

Written elementwise,

$$X_{i,j} = z_{i+j-1}, \qquad 1 \leq i \leq L,\ 1 \leq j \leq K. \tag{15}$$

Therefore, all entries with the same value of $i + j$ are identical, and $\boldsymbol{X}$ has a Hankel structure.

A.1.2 SINGULAR-VALUE DECOMPOSITION AND EIGENTRIPLES

Singular Value Decomposition (SVD) is used to decompose the trajectory matrix.

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}} = \sum_{q=1}^{d} \sigma_q \boldsymbol{u}_q \boldsymbol{v}_q^{\mathsf{T}} \tag{16}$$

where,

$$d = \text{rank}(\boldsymbol{X}), \tag{17}$$

the singular values satisfy

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d > 0, \tag{18}$$

and

$$\boldsymbol{u}_q \in \mathbb{R}^L, \qquad \boldsymbol{v}_q \in \mathbb{R}^K \tag{19}$$

represent the corresponding left and right singular vectors. Each triplet $((\sigma_q, \boldsymbol{u}_q, \boldsymbol{v}_q))$ denotes an SSA eigentriple. Its elementary matrix is:

$$\boldsymbol{X}_q = \sigma_q \boldsymbol{u}_q \boldsymbol{v}_q^{\mathsf{T}}. \tag{20}$$

Hence,

$$\boldsymbol{X} = \sum_{q=1}^{d} \boldsymbol{X}_q. \tag{21}$$

Let

$$\mathcal{G}_{\mathrm{L}} \subseteq \{1, \ldots, d\} \tag{22}$$

be the eigentriple set retained for the construction of the low-variation reconstruction. In the implementation discussed in the paper, $\mathcal{G}_{\mathrm{L}}$ is formed by a specified number of the leading eigentriples. The grouped low-rank trajectory matrix is given by:

$$\boldsymbol{X}_{\mathrm{L}} = \sum_{q \in \mathcal{G}_{\mathrm{L}}} \boldsymbol{X}_q. \tag{23}$$

Thus, the complementary high-variation trajectory matrix is:

$$\boldsymbol{X}_{\mathrm{H}} = \boldsymbol{X} - \boldsymbol{X}_{\mathrm{L}}. \tag{24}$$

Therefore,

$$\boldsymbol{X} = \boldsymbol{X}_{\mathrm{L}} + \boldsymbol{X}_{\mathrm{H}}. \tag{25}$$

The dominant low-rank spatial structure represented by the trajectory matrix is captured by the leading eigentriples, and the residual captures the remaining variation. We refer to the retained versus residual SSA groupings as low-variation and high-variation, respectively, in this work. They do not require an ideal Fourier-domain frequency partition.

A.1.3 DIAGONAL AVERAGING

The grouped matrices in equation equation 23 and equation equation 24 are again mapped to one-dimensional signal space using diagonal averaging (Hankelisation).

For any matrix $\boldsymbol{Y} \in \mathbb{R}^{L\times K}$, we define the set of matrix entries contributing to sample t as:

$$\mathcal{A}_t = \{(i,j) : 1 \le i \le L,\ 1 \le j \le K,\ i+j-1 = t\}\,. \tag{26}$$

The diagonal-averaging operator $\mathcal{H} : \mathbb{R}^{L\times K} \to \mathbb{R}^T$ is defined by:

$$[\mathcal{H}(\boldsymbol{Y})]_t = \frac{1}{|\mathcal{A}_t|} \sum_{(i,j)\in\mathcal{A}_t} Y_{i,j}, \qquad t = 1,\dots,T \tag{27}$$

This compensates for the unequal number of overlapping windows, which contain boundary and interior samples.

The low-variation reconstruction is

$$\boldsymbol{z}_{\mathrm{L}} = \mathcal{H}(\boldsymbol{X}_{\mathrm{L}}), \tag{28}$$

and the high-variation component is

$$\boldsymbol{z}_{\mathrm{H}} = \mathcal{H}(\boldsymbol{X}_{\mathrm{H}}). \tag{29}$$

Since diagonal averaging is linear,

$$\begin{aligned} \boldsymbol{z}_{\mathrm{L}} + \boldsymbol{z}_{\mathrm{H}} &= \mathcal{H}(\boldsymbol{X}_{\mathrm{L}}) + \mathcal{H}(\boldsymbol{X}_{\mathrm{H}}) \\ &= \mathcal{H}(\boldsymbol{X}) \\ &= \boldsymbol{z}. \end{aligned} \tag{30}$$

Therefore, the decomposition is additive in exact arithmetic.

$$\boldsymbol{z} = \boldsymbol{z}_{\mathrm{L}} + \boldsymbol{z}_{\mathrm{H}}. \tag{31}$$

A.1.4 OPERATOR NOTATION

The complete SSA operation for low-variation is written as,

$$\mathcal{S}_{\mathrm{L}}(\boldsymbol{z}; L, \mathcal{G}_{\mathrm{L}}) = \mathcal{H}\left(\sum_{q\in\mathcal{G}_{\mathrm{L}}} \sigma_q \boldsymbol{u}_q \boldsymbol{v}_q^{\mathsf{T}}\right). \tag{32}$$

The complementary operation is

$$\mathcal{S}_{\mathrm{H}}(\boldsymbol{z}; L, \mathcal{G}_{\mathrm{L}}) = \boldsymbol{z} - \mathcal{S}_{\mathrm{L}}(\boldsymbol{z}; L, \mathcal{G}_{\mathrm{L}}). \tag{33}$$

Accordingly,

$$\boldsymbol{z} = \mathcal{S}_{\mathrm{L}}(\boldsymbol{z}) + \mathcal{S}_{\mathrm{H}}(\boldsymbol{z}). \tag{34}$$

In the separable two-dimensional construction, these operators are applied independently along rows and columns of an image, and it is described in Appendix A.2

Algorithm 1 One-dimensional SSA low/high decomposition

Require: Signal $\boldsymbol{z} \in \mathbb{R}^T$
Require: Window length L
Require: Number of retained eigentriples r
Ensure: Low- and high-variation components $\boldsymbol{z}_{\mathrm{L}}, \boldsymbol{z}_{\mathrm{H}}$
1: $K \leftarrow T - L + 1$
2: Construct the trajectory matrix $\boldsymbol{X} \in \mathbb{R}^{L \times K}$ using $X_{i,j} = z_{i+j-1}$
3: Compute $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$
4: $\boldsymbol{X}_{\mathrm{L}} \leftarrow \sum_{q=1}^{r} \sigma_q \boldsymbol{u}_q \boldsymbol{v}_q^{\mathsf{T}}$
5: $\boldsymbol{z}_{\mathrm{L}} \leftarrow \mathcal{H}(\boldsymbol{X}_{\mathrm{L}})$
6: $\boldsymbol{z}_{\mathrm{H}} \leftarrow \boldsymbol{z} - \boldsymbol{z}_{\mathrm{L}}$
7: **return** $\boldsymbol{z}_{\mathrm{L}}, \boldsymbol{z}_{\mathrm{H}}$

A.1.5 Application to multichannel signals

Consider a signal with C output channels ($Z \in \mathbb{R}^{T \times C}$). SSA is applied independently to each channel:

$$\boldsymbol{z}_{\mathrm{L}}^{(c)} = \mathcal{S}_{\mathrm{L}} \left(\boldsymbol{z}^{(c)}; L, \mathcal{G}_{\mathrm{L}} \right), \tag{35}$$

$$\boldsymbol{z}_{\mathrm{H}}^{(c)} = \boldsymbol{z}^{(c)} - \boldsymbol{z}_{\mathrm{L}}^{(c)}, \qquad c = 1, \ldots, C \tag{36}$$

Then, the channel-wise outputs are stacked. This results in $Z_{\mathrm{L}}, Z_{\mathrm{H}} \in \mathbb{R}^{T \times C}$.

For an RGB image, $C = 3$, and the same window length and retained-rank setting are used for the three colour channels at a given decomposition stage.

A.1.6 Algorithmic summary

Algorithm 1 summarises the one-dimensional operation.

A.1.7 Role within the proposed framework

In the proposed framework, SSA is used to obtain branch-specific targets to define branch responsibilities, not as the final representation model. For image representation, the one-dimensional formulation of SSA is applied first along the horizontal coordinate and then along the vertical coordinate. This results in four targets, $I_{\mathrm{S}}^{\mathrm{SSA}}$, $I_{\mathrm{X}}^{\mathrm{SSA}}$, $I_{\mathrm{Y}}^{\mathrm{SSA}}$, and $I_{\mathrm{D}}^{\mathrm{SSA}}$.

The gated contribution of each branch is supervised by its corresponding SSA targets during the branch pretraining phase, and then it is weakly supervised with decayed SSA loss during the joint training phase.

A.2 Separable Two-Dimensional SSA Target Construction

This appendix contains the complete construction of the two-dimensional separable SSA used for branch-specific target construction. Here, we apply one-dimensional SSA along two coordinate axes independently. At each stage, SSA decomposes the input signal into low- and high-variation components using retained and residual eigentriples. Applying this binary decomposition along two coordinate axes provide 4 additive components.

A.2.1 Notation and one-dimensional SSA operator

Let $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^C$ be an image with spatial coordinates $\boldsymbol{x} = (x, y)$, where $C = 3$ for an RGB image. The decomposition is applied independently to each channel.

For a given spatial axis $d \in \{x, y\}$, let $\mathcal{S}_{\mathrm{L}}^{(d)}[Z]$ be the low-variation reconstruction obtained by applying one-dimensional SSA along axis d. Then, the corresponding high-dimensional component is given by the residual:

$$\mathcal{S}_{\mathrm{H}}^{(d)}[Z] = Z - \mathcal{S}_{\mathrm{L}}^{(d)}[Z]. \tag{37}$$

This makes the axis-wise decomposition exactly additive.

$$Z = \mathcal{S}_{\mathrm{L}}^{(d)}[Z] + \mathcal{S}_{\mathrm{H}}^{(d)}[Z]. \tag{38}$$

The trajectory-matrix construction, eigentriple selection, low-rank reconstruction, and diagonal averaging used inside $\mathcal{S}_{\mathrm{L}}^{(d)}$ are given in Appendix A.1.

A.2.2 First-stage decomposition along the horizontal coordinate

First, SSA is applied along the horizontal direction. For each fixed vertical coordinate and channel, the corresponding horizontal row in the image is treated as a one-dimensional signal. This produces:

$$I_{x\mathrm{L}} = \mathcal{S}_{\mathrm{L}}^{(x)}[I], \tag{39}$$

$$I_{x\mathrm{H}} = \mathcal{S}_{\mathrm{H}}^{(x)}[I] = I - I_{x\mathrm{L}}. \tag{40}$$

Hence,

$$I = I_{x\mathrm{L}} + I_{x\mathrm{H}}. \tag{41}$$

The subscript $x\mathrm{L}$ denotes that the component was obtained by considering the low-frequency variation along the horizontal axis, whereas the subscript $x\mathrm{H}$ indicates the complementary high-frequency component. The first symbol denotes the direction along which one-dimensional SSA was applied, and the second symbol indicates the corresponding variation range.

A.2.3 Second-stage decomposition along the vertical coordinate

Then, the resulting two components are decomposed independently along the vertical coordinate axis y. Applying the vertical low/high split to $I_{x\mathrm{L}}$ gives:

$$I_{\mathrm{LL}} = \mathcal{S}_{\mathrm{L}}^{(y)}[I_{x\mathrm{L}}], \tag{42}$$

$$I_{\mathrm{LH}} = \mathcal{S}_{\mathrm{H}}^{(y)}[I_{x\mathrm{L}}] = I_{x\mathrm{L}} - I_{\mathrm{LL}} \tag{43}$$

Similarly, decomposing $I_{x\mathrm{H}}$ along y gives:

$$I_{\mathrm{HL}} = \mathcal{S}_{\mathrm{L}}^{(y)}[I_{x\mathrm{H}}], \tag{44}$$

$$I_{\mathrm{HH}} = \mathcal{S}_{\mathrm{H}}^{(y)}[I_{x\mathrm{H}}] = I_{x\mathrm{H}} - I_{\mathrm{HL}} \tag{45}$$

Under this notation, the first symbol denotes the variation range along the horizontal axis, while the second subscript denotes the variation range along the vertical axis.

Therefore,

- I_{LL} is low variation along both coordinates;
- I_{HL} is high variation along x and low variation along y;
- I_{LH} is low variation along x and high variation along y; and
- I_{HH} is high variation along both coordinates.

Using the additivity of both stages,

$$\begin{aligned} I &= I_{x\mathrm{L}} + I_{x\mathrm{H}} \\ &= (I_{\mathrm{LL}} + I_{\mathrm{LH}}) + (I_{\mathrm{HL}} + I_{\mathrm{HH}}) \\ &= I_{\mathrm{LL}} + I_{\mathrm{HL}} + I_{\mathrm{LH}} + I_{\mathrm{HH}}. \end{aligned} \tag{46}$$

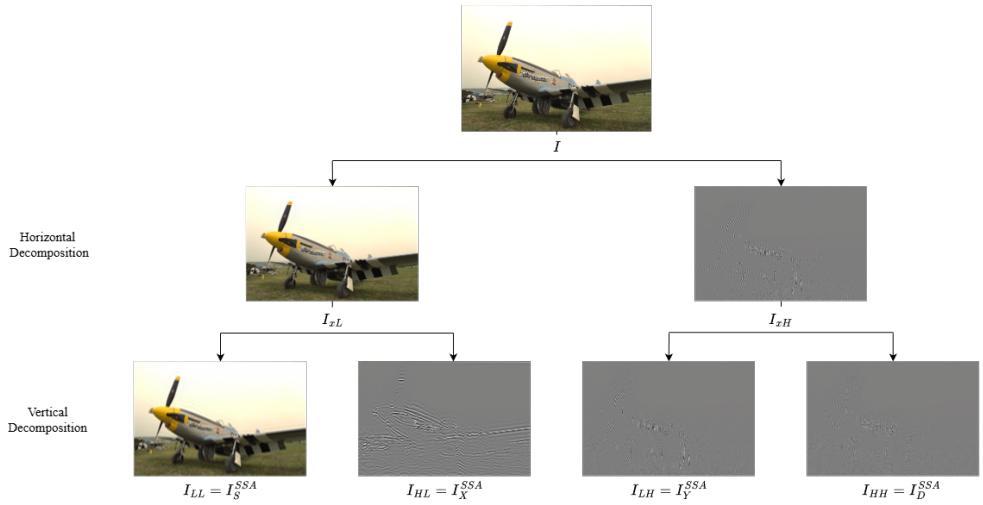


Figure 5: Visual summary of the separable two-dimensional SSA target construction. The original image is first decomposed along the horizontal coordinate into the low-variation component $I_{x\mathrm{L}}$ and the high-variation residual $I_{x\mathrm{H}}$. Each intermediate component is then decomposed along the vertical coordinate, producing I_{LL}, I_{HL}, I_{LH}, and I_{HH}. These are relabelled as the structural, horizontal-detail, vertical-detail, and joint-detail targets $I_{\mathrm{S}}^{\mathrm{SSA}}$, $I_{\mathrm{X}}^{\mathrm{SSA}}$, $I_{\mathrm{Y}}^{\mathrm{SSA}}$, and $I_{\mathrm{D}}^{\mathrm{SSA}}$, respectively. Detail components are displayed with a shifted or contrast-enhanced visualisation only for readability; the numerical targets used during training retain their signed values.

A.2.4 Mapping the SSA components to branch roles

The four separable components are relabelled according to their functional roles in the multi-branch INR:

$$I_{\mathrm{S}}^{\mathrm{SSA}} = I_{\mathrm{LL}}, \qquad I_{\mathrm{X}}^{\mathrm{SSA}} = I_{\mathrm{HL}}, \qquad I_{\mathrm{Y}}^{\mathrm{SSA}} = I_{\mathrm{LH}}, \qquad I_{\mathrm{D}}^{\mathrm{SSA}} = I_{\mathrm{HH}} \tag{47}$$

Here, S denotes the smooth structural component, X contains the high-variation components along the horizontal direction, Y contains the high-variation components along the vertical direction, and D contains the joint high-variation component.

Thus, the branch target set is given by:

$$\left\{I_b^{\mathrm{SSA}}\right\}_{b\in\mathcal{B}} = \left\{I_{\mathrm{S}}^{\mathrm{SSA}}, I_{\mathrm{X}}^{\mathrm{SSA}}, I_{\mathrm{Y}}^{\mathrm{SSA}}, I_{\mathrm{D}}^{\mathrm{SSA}}\right\}, \qquad \mathcal{B} = \{\mathrm{S},\mathrm{X},\mathrm{Y},\mathrm{D}\}. \tag{48}$$

Each branch is trained against the SSA component with the same role label. Since the four targets preserve additivity as given in the equation equation 46, their gated branch contributions can be combined using additive fusion.

A.2.5 Implementation procedure

The target construction is summarised in algorithm 2. The operation is separately applied to every colour channel. Horizontal and vertical decompositions may use different window sizes and retained eigentriples, according to the image experiments reported in the main paper.

A.2.6 Axis order and interpretation

The decomposition in the implementation was first performed along the horizontal axis, and it was followed by the vertical decomposition. Since the second SSA decomposition uses the components obtained in the first stage, this order should be preserved. The fixed order ensures a valid additive decomposition and a consistent set of branch roles. The labels X and Y denote the coordinate along which a given component retains high variation. It does not indicate an orientation of an edge in the displayed image.

Algorithm 2 Separable two-dimensional SSA target construction

Require: Image $I \in \mathbb{R}^{H\times W\times C}$
Require: Horizontal SSA parameters (L_x, r_x)
Require: Vertical SSA parameters (L_y, r_y)
Ensure: Targets $\{I_{\mathrm{S}}^{\mathrm{SSA}}, I_{\mathrm{X}}^{\mathrm{SSA}}, I_{\mathrm{Y}}^{\mathrm{SSA}}, I_{\mathrm{D}}^{\mathrm{SSA}}\}$

$I_{x\mathrm{L}} \leftarrow \mathcal{S}_{\mathrm{L}}^{(x)}(I; L_x, r_x)$
$I_{x\mathrm{H}} \leftarrow I - I_{x\mathrm{L}}$
$I_{\mathrm{LL}} \leftarrow \mathcal{S}_{\mathrm{L}}^{(y)}(I_{x\mathrm{L}}; L_y, r_y)$
$I_{\mathrm{LH}} \leftarrow I_{x\mathrm{L}} - I_{\mathrm{LL}}$
$I_{\mathrm{HL}} \leftarrow \mathcal{S}_{\mathrm{L}}^{(y)}(I_{x\mathrm{H}}; L_y, r_y)$
$I_{\mathrm{HH}} \leftarrow I_{x\mathrm{H}} - I_{\mathrm{HL}}$
$I_{\mathrm{S}}^{\mathrm{SSA}} \leftarrow I_{\mathrm{LL}}$
$I_{\mathrm{X}}^{\mathrm{SSA}} \leftarrow I_{\mathrm{HL}}$
$I_{\mathrm{Y}}^{\mathrm{SSA}} \leftarrow I_{\mathrm{LH}}$
$I_{\mathrm{D}}^{\mathrm{SSA}} \leftarrow I_{\mathrm{HH}}$
return $\{I_{\mathrm{S}}^{\mathrm{SSA}}, I_{\mathrm{X}}^{\mathrm{SSA}}, I_{\mathrm{Y}}^{\mathrm{SSA}}, I_{\mathrm{D}}^{\mathrm{SSA}}\}$

The decomposition does not claim that it contains a unique physical partition of the image. Instead, it is used as a data-dependent task allocation mechanism. During the branch pretraining, each branch is optimised to approximate the corresponding target obtained by SSA decomposition, whereas in the joint training, reconstruction loss drives the model to refine the final reconstruction while the decaying SSA loss provides a weak supervision to preserve the initial division of responsibilities.

A.3 SSA-Derived Gate Priors

This appendix describes the construction of SSA-derived gate priors used to guide the strength of each specialised branch's contribution spatially. While SSA targets determine *what* each branch should represent, gate priors provide a weak indication of *how strongly* a certain branch should contribute at a given spatial location. The magnitudes of the corresponding SSA targets are used to obtain the priors. These priors are used as independent role indicators rather than as a mutually exclusive set of weights.

A.3.1 Branch activity from SSA targets

Let

$$I_b^{\mathrm{SSA}}(\boldsymbol{x}) \in \mathbb{R}^C, \qquad b \in \mathcal{B} \tag{49}$$

be the SSA target of branch b, evaluated at coordinate $\boldsymbol{x}$. For an image $\mathcal{B} = \{\mathrm{S}, \mathrm{X}, \mathrm{Y}, \mathrm{D}\}$, and for RGB images, $C = 3$.

First, the target magnitudes are used to compute a scalar activity value.

$$A_b(\boldsymbol{x}) = \frac{1}{C}\sum_{c=1}^{C} \left| I_b^{\mathrm{SSA}}(\boldsymbol{x}, c) \right| \tag{50}$$

For RGB images, this becomes

$$A_b(\boldsymbol{x}) = \frac{1}{3}\sum_{c=1}^{3} \left| I_b^{\mathrm{SSA}}(\boldsymbol{x}, c) \right| \tag{51}$$

The absolute value is used because the detail components produced by SSA are signed residuals. Both positive and negative values indicate branch activity. Therefore, allowing negative values might

cause cancellation between channels or between opposite local variations. Averaging across channels results in one scalar prior per branch and coordinate, matching the single scalar gate predicted by each branch.

A.3.2 ACROSS-BRANCH NORMALISATION

A normalisation is performed for branch activities relative to the strongest SSA component at the same coordinate.

$$\widetilde{A}_b(\boldsymbol{x}) = \frac{A_b(\boldsymbol{x})}{\max_{j\in\mathcal{B}} A_j(\boldsymbol{x}) + \epsilon} \tag{52}$$

where $\epsilon > 0$ is a small numerical stabiliser.

This normalisation step has two useful properties.

1. It expresses branch activities relative to the most active component at a given coordinate, removing the effects of the absolute local amplitude scale.
2. It preserves independent branch activity since more than one branch may have large priors when multiple SSA components are strong.

The normalisation in equation equation 52 is not a probability normalisation. Therefore, it does not impose the sum-to-one constraint.

$$\sum_{b\in\mathcal{B}} \widetilde{A}_b(\boldsymbol{x}) \neq 1 \quad \text{in general} \tag{53}$$

This aligns with the additive structure of the SSA decomposition, in which more than one component may contribute at the same spatial location.

A.3.3 CONTRAST SHAPING

The final SSA-derived gate prior is obtained by applying a contrast exponent:

$$G_b^{\mathrm{SSA}}(\boldsymbol{x}) = \left(\widetilde{A}_b(\boldsymbol{x})\right)^{\gamma}, \qquad \gamma > 0. \tag{54}$$

Combining

equations equation 50–equation 54 gives

$$G_b^{\mathrm{SSA}}(\boldsymbol{x}) = \left(\frac{\frac{1}{C}\sum_{c=1}^{C} \left| I_b^{\mathrm{SSA}}(\boldsymbol{x}, c) \right|}{\max_{j\in\mathcal{B}} \left(\frac{1}{C}\sum_{c=1}^{C} \left| I_j^{\mathrm{SSA}}(\boldsymbol{x}, c) \right| \right) + \epsilon} \right)^{\gamma} \tag{55}$$

Here, the exponent γ controls the contrast of the prior maps. $0 < \gamma < 1$ produces soft and spatially extended priors by increasing moderate activity values, whereas $\gamma > 1$ makes priors more selective by suppressing weaker activities. When $\gamma = 1$, the normalised activities are used directly. In the image experiments, the main model uses $\gamma = 0.75$ which provides soft guidance while retaining low-amplitude detail regions.

The resulting prior vector is

$$\boldsymbol{G}^{\mathrm{SSA}}(\boldsymbol{x}) = \left[G_{\mathrm{S}}^{\mathrm{SSA}}(\boldsymbol{x}), G_{\mathrm{X}}^{\mathrm{SSA}}(\boldsymbol{x}), G_{\mathrm{Y}}^{\mathrm{SSA}}(\boldsymbol{x}), G_{\mathrm{D}}^{\mathrm{SSA}}(\boldsymbol{x})\right] \in [0,1]^4 \tag{56}$$

A.3.4 LOW-ENERGY COORDINATES

The implementation uses $\epsilon = 10^{-8}$. Let

$$A_{\max}(\boldsymbol{x}) = \max_{j \in \mathcal{B}} A_j(\boldsymbol{x}). \tag{57}$$

Before the fallback rule, coordinates satisfying $A_{\max}(\boldsymbol{x}) < \epsilon$ produces gate targets close to zero for all branches. In the implementation, we assign such coordinates to the structural branch.

$$G_{\mathrm{S}}^{\mathrm{SSA}}(\boldsymbol{x}) \leftarrow 1, \qquad G_b^{\mathrm{SSA}}(\boldsymbol{x}) \leftarrow 0 \ \text{ for } b \neq \mathrm{S}, \qquad \text{if } A_{\max}(\boldsymbol{x}) < \epsilon. \tag{58}$$

This prevents numerically inactive pixels from assigning zero targets to every gate. It does not modify the SSA targets.

A.3.5 RELATION TO LEARNED GATES

Each branch predicts a gate logit $\alpha_b(\boldsymbol{x})$, which is then converted to an independent sigmoid gate.

$$g_b(\boldsymbol{x}) = g_{\min} + (1 - g_{\min})\sigma\left(\frac{\alpha_b(\boldsymbol{x})}{\tau}\right) \tag{59}$$

Here, g_{min} denotes the gate floor, and τ is the gate temperature.

The branch contribution is given by:

$$\boldsymbol{c}_b(\boldsymbol{x}) = g_b(\boldsymbol{x})\boldsymbol{y}_b(\boldsymbol{x}) \tag{60}$$

The gate prior and the gate floor have different responsibilities. While G_b^{SSA} defines a spatial activity target derived from the corresponding SSA component, $g_{\min}$ prevents a branch being completely suppressed ensuring a minimum gradient path. Thus, at convergence, the learned gate is not expected to reproduce the exact prior. Instead, the prior defines the initial division of responsibilities over the space, while during the joint training, the gates adapt to the final reconstruction objective.

A.3.6 GATE-PRIOR SUPERVISION

During branch-specialised training, branch b is guided using:

$$\mathcal{L}_{\mathrm{pg},b} = \frac{1}{|\Omega_{\mathrm{d}}|} \sum_{\boldsymbol{x} \in \Omega_{\mathrm{d}}} \left| g_b(\boldsymbol{x}) - G_b^{\mathrm{SSA}}(\boldsymbol{x}) \right|^2 \tag{61}$$

The branch-specialised objective is

$$\mathcal{L}_{\mathrm{spec},b} = \frac{1}{|\Omega_{\mathrm{d}}|} \sum_{\boldsymbol{x} \in \Omega_{\mathrm{d}}} \left\| \boldsymbol{c}_b(\boldsymbol{x}) - I_b^{\mathrm{SSA}}(\boldsymbol{x}) \right\|_2^2 + \lambda_{\mathrm{pg}} \mathcal{L}_{\mathrm{pg},b} \tag{62}$$

In the joint training phase, all gate priors are supervised together:

$$\mathcal{L}_{\mathrm{gate}} = \frac{1}{|\Omega_{\mathrm{d}}|\,|\mathcal{B}|} \sum_{\boldsymbol{x} \in \Omega_{\mathrm{d}}} \sum_{b \in \mathcal{B}} \left| g_b(\boldsymbol{x}) - G_b^{\mathrm{SSA}}(\boldsymbol{x}) \right|^2 \tag{63}$$

where

$$\boldsymbol{G}(\boldsymbol{x}) = [g_{\mathrm{S}}(\boldsymbol{x}), g_{\mathrm{X}}(\boldsymbol{x}), g_{\mathrm{Y}}(\boldsymbol{x}), g_{\mathrm{D}}(\boldsymbol{x})] \tag{64}$$

The weight of the gate prior term $\lambda_g(t)$ is decayed during training, making it a weak supervision over time.

Algorithm 3 SSA-derived gate-prior construction

Require: SSA targets $\{I_b^{\mathrm{SSA}}\}_{b\in\mathcal{B}}$, each in $\mathbb{R}^{H\times W\times C}$
Require: Contrast exponent γ
Require: Numerical stabiliser ϵ
Require: Numerical zero-energy threshold equal to ϵ
Ensure: Gate priors $\{G_b^{\mathrm{SSA}}\}_{b\in\mathcal{B}}$
for each branch $b \in \mathcal{B}$ **do**
  $A_b \leftarrow \frac{1}{C}\sum_{c=1}^{C} |I_b^{\mathrm{SSA}}(:,:,c)|$
end for
$A_{\max} \leftarrow \max_{j\in\mathcal{B}} A_j$
for each branch $b \in \mathcal{B}$ **do**
  $G_b^{\mathrm{SSA}} \leftarrow [\mathrm{clip}\,(A_b/(A_{\max}+\epsilon), 0, 1)]^{\gamma}$
end for
$\mathcal{M}_{\mathrm{zero}} \leftarrow \{\, \boldsymbol{x} : A_{\max}(\boldsymbol{x}) < \epsilon \,\}$
if $\mathcal{M}_{\mathrm{zero}} \neq \varnothing$ **then**
  $G_{\mathrm{S}}^{\mathrm{SSA}}(\mathcal{M}_{\mathrm{zero}}) \leftarrow 1$
  for each $b \in \mathcal{B} \setminus \{\mathrm{S}\}$ **do**
    $G_b^{\mathrm{SSA}}(\mathcal{M}_{\mathrm{zero}}) \leftarrow 0$
  end for
end if
return $\{G_b^{\mathrm{SSA}}\}_{b\in\mathcal{B}}$

A.3.7 Why independent priors are used

A softmax gate imposes the sum-to-one condition on the learned gate values. That is,

$$\sum_{b\in\mathcal{B}} g_b(\boldsymbol{x}) = 1 \tag{65}$$

This forces branches to compete for a fixed amount of activity, making it inconsistent with the additive model:

$$I(\boldsymbol{x}) = \sum_{b\in\mathcal{B}} I_b^{\mathrm{SSA}}(\boldsymbol{x}) \tag{66}$$

Independent priors and independent sigmoid gates allow the model to make any combination of branch contributions without suppressing other branches. Therefore, the role of the priors is to encourage each branch to contribute at spatial locations where its corresponding SSA component is active.

A.3.8 Algorithmic summary

Algorithm 3 summarises the gate-prior construction.

A.3.9 Implementation settings

For the experiments on images discussed in the main paper, the gate prior contrast is set to $\gamma = 0.75$. The learned gate temperature is $\tau = 1.0$ and the gate floor is $g_{min} = 0.02$. The branch-specific gate prior weight is $\lambda_{\mathrm{pg}} = 5 \times 10^{-4}$. During the joint training, the gate prior weight is decayed from 2×10^{-4} to zero.

A.4 Generalisation to N-Dimensional Signals

The main paper discusses the methodology for an Image, and assigns four parallel branches to the structural, horizontal-detail, vertical-detail, and joint-detail components produced by SSA decomposition. This appendix extends the same idea to an N-dimensional signal. The generalisation preserves the same properties of the proposed method.

- A seperable SSA decomposition along signal coordinates and two frequency ranges.
- Explicit branch target for all parallel branches
- Independent gate additive fusion
- Branch-specific learnable coordinate scaling

A.4.1 SIGNAL MODEL AND BRANCH INDEX SET

Let $I : \Omega \subset \mathbb{R}^N \to \mathbb{R}^C$ be an N-dimensional signal with C output channels, and $\boldsymbol{x} = (x_1, \ldots, x_N) \in [-1, 1]^N$ denotes its normalised coordinate vector.

For the simplest implementation, we assign L or H label each dimension. The role of a given component is therefore denoted by:

$$\boldsymbol{r} = (r_1, \ldots, r_N) \in \{\mathrm{L}, \mathrm{H}\}^N \tag{67}$$

Thus, the complete branch set is

$$\mathcal{B}_N = \{\mathrm{L}, \mathrm{H}\}^N, \qquad |\mathcal{B}_N| = 2^N. \tag{68}$$

Or, a binary index $\boldsymbol{r} \in \{0, 1\}^N$ can be used, where 0 denotes low-variation while 1 denotes high-variation.

A.4.2 SEPARABLE N-DIMENSIONAL SSA DECOMPOSITION

For each dimension $n \in \{1, \ldots, N\}$, let

$$\mathcal{S}_{\mathrm{L}}^{(n)} \quad \text{and} \quad \mathcal{S}_{\mathrm{H}}^{(n)} \tag{69}$$

denote he one-dimensional SSA low- and high-variation reconstruction operations applied along dimension N independently over all fibres formed with remaining coordinates and output channel kept fixed. The high-variation operation is given by the residual:

$$\mathcal{S}_{\mathrm{H}}^{(n)}[Z] = Z - \mathcal{S}_{\mathrm{L}}^{(n)}[Z], \tag{70}$$

so that

$$Z = \mathcal{S}_{\mathrm{L}}^{(n)}[Z] + \mathcal{S}_{\mathrm{H}}^{(n)}[Z] \tag{71}$$

for every intermediate signal Z.

Since SSA is applied sequentially, the axis order $\pi = (\pi_1, \ldots, \pi_N)$ is preserved. We start from $I^{(0)} = I$, and every intermediate component is split along the next dimension in sequence. For a role $\boldsymbol{r} = (r_1, \ldots, r_N)$, the resulting component target is given by:

$$I_{\boldsymbol{r}}^{\mathrm{SSA}} = \mathcal{S}_{r_{\pi_N}}^{(\pi_N)} \circ \mathcal{S}_{r_{\pi_{N-1}}}^{(\pi_{N-1})} \circ \cdots \circ \mathcal{S}_{r_{\pi_1}}^{(\pi_1)}[I] \tag{72}$$

The operators given in equation equation 72 are recursively applied to the intermediate components. Therefore, we treat the order of decomposition as a part of the SSA target construction.

Repeated use of the additive low-/high split gives

$$I = \sum_{\boldsymbol{r} \in \mathcal{B}_N} I_{\boldsymbol{r}}^{\mathrm{SSA}}. \tag{73}$$

Therefore, all the components correspond to one combination of variation ranges across the N signal dimensions. The role with all low frequencies $\boldsymbol{r}_{\mathrm{S}} = (\mathrm{L}, \ldots, \mathrm{L})$ represents the structural component, whereas the roles containing a combination of high-frequencies represent directional

or joint high-variation components. A component with only one high-frequency entry represents variations primarily along that dimension, while roles with several high-frequency entries model joint variations along those dimensions.

For $N = 2$, the role set becomes

$$\mathcal{B}_2 = \{(\mathrm{L}, \mathrm{L}), (\mathrm{H}, \mathrm{L}), (\mathrm{L}, \mathrm{H}), (\mathrm{H}, \mathrm{H})\} \tag{74}$$

These roles correspond with S, X, Y, and D branches mentioned in the main paper.

A.4.3 Role-defined coordinate-network branches

All retained roles $bmr \in \mathcal{B}_N$ is assigned a coordinate network $F_{\boldsymbol{r}} : \mathbb{R}^N \to \mathbb{R}^{C+1}$.

The branch receives a scaled coordinate as input

$$\widetilde{\boldsymbol{x}}_{\boldsymbol{r}} = \boldsymbol{s}_{\boldsymbol{r}} \odot \boldsymbol{x}, \qquad \boldsymbol{s}_{\boldsymbol{r}} \in \mathbb{R}^N_{>0}, \tag{75}$$

and predicts a component with C channels and a scalar gate logit.

$$F_{\boldsymbol{r}}(\widetilde{\boldsymbol{x}}_{\boldsymbol{r}}) = [\boldsymbol{y}_{\boldsymbol{r}}(\boldsymbol{x}), \alpha_{\boldsymbol{r}}(\boldsymbol{x})] \tag{76}$$

where

$$\boldsymbol{y}_{\boldsymbol{r}}(\boldsymbol{x}) \in \mathbb{R}^C, \qquad \alpha_{\boldsymbol{r}}(\boldsymbol{x}) \in \mathbb{R}. \tag{77}$$

A direct role-aligned scale initialisation is

$$s^{(0)}_{\boldsymbol{r},n} = \begin{cases} 1, & r_n = \mathrm{L}, \\ m_n, & r_n = \mathrm{H}, \end{cases} \tag{78}$$

Here, $m_n > 0$ is the high-frequency multiplier for dimension n. With this initialisation choice, a branch is exposed to a higher effective coordinate frequency along every dimension labelled as H. This input scale is kept learnable through the log-space parametrisation:

$$\boldsymbol{s}_{\boldsymbol{r}} = \exp(\boldsymbol{\eta}_{\boldsymbol{r}}), \tag{79}$$

This preserves positivity during optimisation.

A.4.4 Independent gates and additive fusion

The gate of branch $\boldsymbol{r}$ is

$$g_{\boldsymbol{r}}(\boldsymbol{x}) = g_{\min} + (1 - g_{\min})\sigma\left(\frac{\alpha_{\boldsymbol{r}}(\boldsymbol{x})}{\tau}\right), \tag{80}$$

and its effective contribution is given by:

$$\boldsymbol{c}_{\boldsymbol{r}}(\boldsymbol{x}) = g_{\boldsymbol{r}}(\boldsymbol{x})\boldsymbol{y}_{\boldsymbol{r}}(\boldsymbol{x}) \tag{81}$$

Additive fusion is used to obtain the final reconstruction

$$\widehat{I}(\boldsymbol{x}) = \sum_{\boldsymbol{r} \in \mathcal{B}_N} \boldsymbol{c}_{\boldsymbol{r}}(\boldsymbol{x}). \tag{82}$$

Independent sigmoid gates are used due to the additivity of the SSA components derived in equation equation 73. And since sigmoid gates does not impose the sum-to-one constraint, several component roles are allowed to be active at the same coordinate.

A.4.5 GENERALISED SSA-DERIVED GATE PRIORS

For each role $\boldsymbol{r}$, a scalar activity map is defined from the corresponding SSA target:

$$A_{\boldsymbol{r}}(\boldsymbol{x}) = \frac{1}{C}\sum_{c=1}^{C}\left|I_{\boldsymbol{r}}^{\mathrm{SSA}}(\boldsymbol{x}, c)\right| \tag{83}$$

The generalised gate prior is given by:

$$G_{\boldsymbol{r}}^{\mathrm{SSA}}(\boldsymbol{x}) = \left(\frac{A_{\boldsymbol{r}}(\boldsymbol{x})}{\max_{\boldsymbol{q}\in\mathcal{B}_N} A_{\boldsymbol{q}}(\boldsymbol{x}) + \epsilon}\right)^{\gamma} \tag{84}$$

Similar to the two-dimensional formulation, these priors act as independent activity indicators without the sum-to-one constraint.

$$\sum_{\boldsymbol{r}\in\mathcal{B}_N} G_{\boldsymbol{r}}^{\mathrm{SSA}}(\boldsymbol{x}) \neq 1 \quad \text{in general.} \tag{85}$$

A.4.6 TRAINING OBJECTIVE

Let Ω_{d} denote the sampled discrete coordinate set. The reconstruction loss is defined as:

$$\mathcal{L}_{\mathrm{rec}} = \frac{1}{|\Omega_{\mathrm{d}}|}\sum_{\boldsymbol{x}\in\Omega_{\mathrm{d}}}\left\|\widehat{I}(\boldsymbol{x}) - I(\boldsymbol{x})\right\|_2^2 \tag{86}$$

Branch accountability is ensured through:

$$\mathcal{L}_{\mathrm{SSA}} = \sum_{\boldsymbol{r}\in\mathcal{B}_N}\frac{1}{|\Omega_{\mathrm{d}}|}\sum_{\boldsymbol{x}\in\Omega_{\mathrm{d}}}\left\|\boldsymbol{c}_{\boldsymbol{r}}(\boldsymbol{x}) - I_{\boldsymbol{r}}^{\mathrm{SSA}}(\boldsymbol{x})\right\|_2^2 \tag{87}$$

The gate-prior loss is

$$\mathcal{L}_{\mathrm{gate}} = \frac{1}{|\Omega_{\mathrm{d}}|\,|\mathcal{B}_N|}\sum_{\boldsymbol{x}\in\Omega_{\mathrm{d}}}\sum_{\boldsymbol{r}\in\mathcal{B}_N}\left|g_{\boldsymbol{r}}(\boldsymbol{x}) - G_{\boldsymbol{r}}^{\mathrm{SSA}}(\boldsymbol{x})\right|^2 \tag{88}$$

and the raw-component amplitude penalty is

$$\mathcal{L}_{\mathrm{amp}} = \frac{1}{|\mathcal{B}_N|}\sum_{\boldsymbol{r}\in\mathcal{B}_N}\mathbb{E}_{\boldsymbol{x}\in\Omega_{\mathrm{d}}}\left[\|\boldsymbol{y}_{\boldsymbol{r}}(\boldsymbol{x})\|_2^2\right] \tag{89}$$

The complete objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{SSA}}(t)\mathcal{L}_{\mathrm{SSA}} + \lambda_g(t)\mathcal{L}_{\mathrm{gate}} + \lambda_a\mathcal{L}_{\mathrm{amp}} \tag{90}$$

Similar to the two-dimensional case given in the main paper, the two-stage training process is applied to any N-dimensional signal.

First, each branch is trained independently using

$$\mathcal{L}_{\mathrm{spec},\boldsymbol{r}} = \frac{1}{|\Omega_{\mathrm{d}}|}\sum_{\boldsymbol{x}\in\Omega_{\mathrm{d}}}\Bigg[\left\|\boldsymbol{c}_{\boldsymbol{r}}(\boldsymbol{x}) - I_{\boldsymbol{r}}^{\mathrm{SSA}}(\boldsymbol{x})\right\|_2^2 + \lambda_{\mathrm{pg}}\left|g_{\boldsymbol{r}}(\boldsymbol{x}) - G_{\boldsymbol{r}}^{\mathrm{SSA}}(\boldsymbol{x})\right|^2\Bigg] \tag{91}$$

Then, all the branches are jointly optimised by the overall objective given by equation equation 90.

A.4.7 PSO INITIALISATION IN N DIMENSIONS

For the complete branch set, all scale parameters can be represented by:

$$\boldsymbol{q} = \left[\boldsymbol{s}_{\boldsymbol{r}_1}^{\mathsf{T}}, \ldots, \boldsymbol{s}_{\boldsymbol{r}_{2^N}}^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{N2^N}. \tag{92}$$

PSO is used to initialise this vector before training using gradient-based optimisation. A short fitting guided by the SSA targets is performed as the fitness evaluation. Since the search dimension grows as $N2^N$, complete PSO might not be practical for higher Ns. Therefore, we may optimise only a subset of shared scale multipliers and initialise through equation 78.

A.4.8 SPECIAL CASES

One-dimensional signals. For $N = 1$, $\mathcal{B}_1 = \{\mathrm{L}, \mathrm{H}\}$ result in two branches. These align with the structural and detail branches used for audio signals.

$$I = I_{\mathrm{L}}^{\mathrm{SSA}} + I_{\mathrm{H}}^{\mathrm{SSA}}, \qquad \widehat{I}(t) = c_{\mathrm{L}}(t) + c_{\mathrm{H}}(t). \tag{93}$$

Two-dimensional signals. For $N = 2$, the four roles are (L, L), (H, L), (L, H), (H, H). This recovers the S, X, Y, and D branches discussed for the image model.

Three-dimensional signals. For $N = 3$, the simplest decomposition contains eight branches. The component containing low frequencies for all dimensions represents the volumetric structure; the components containing a single high frequency represent directional high-variation detail; the components containing high frequencies along two directions contain joint high-variation information along the corresponding dimensions; and the component containing all high frequencies represents joint variation across all three dimensions.

A.5 COMPLETE KODAK RESULTS AND EVALUATION DETAILS

The complete per-image reconstruction results of the proposed SSA-guided multi-branch INR on the 24-image Kodak dataset and clarification of the evaluation and checkpoint-selection procedure used to obtain the values shown in the main paper are reported in this appendix. A machine-readable CSV containing the recorded run configuration and the same per-image results is included with the supplementary material.

A.5.1 IMPLEMENTATION DETAILS

RGB intensities are normalized to $[0, 1]$ and the corresponding two-dimensional coordinates are mapped to $[-1, 1]^2$ for image representation. The 4 branches consist of 3 sinusoidal hidden layers of width 256 followed by a linear four-channel output layer each, with three channels representing the branch component and one channel representing its gate logit. The first sinusoidal layer uses $\omega_0 = 60$ and the following layers use $\omega_0 = 30$.The role aligned coordinate scales of S, X, Y, and D branches are initialized to $(1, 1)$, $(6, 1)$,$(1, 6)$, and $(6, 6)$ respectively.Separable SSA is applied independently to each colour channel using horizontal and vertical window lengths of 64, retaining 16 horizontal and 8 vertical eigentriples for the corresponding low-variation reconstructions. The gate temperature is set to 1.0, and the gate floor and gate-prior are set to 0.02, and 0.75, respectively. Each branch is pre-trained using Adam for 500 epochs independently with a learning rate of 10^{-4}, a coordinate batch size of 8192, and a gate-prior weight of 5×10^{-4}. Afterwards, joint training is conducted using Adam for 500 epochs with an initial learning rate of 10^{-4} and cosine annealing to 10^{-6}. The SAA weight initially decays from 0.03 to 0.003, The weighted contribution of the SSA weight is capped at 25% of the reconstruction loss. The gate-prior weight is decayed from 2×10^{-4} to zero. The raw-component amplitude penalty is weighted by 10^{-8}. All image coordinates are processed each epoch in a randomly shuffled order, and the best checkpoint is selected every 25 epochs according to the full-image reconstruction MSE. PSNR is computed as $-10 \log_{10}(\mathrm{MSE})$ as the image values lie in $[0, 1]$.

Table 3: Reconstruction results of the proposed method on 24 Kodak images. selected full-image checkpoint is used to compute PSNR and MSE. Runtime is the recorded wall-clock time.

Image	PSNR (dB) ↑	MSE ↓	Runtime (min)
kodim01	47.040	1.977e-05	12.11
kodim02	50.239	9.466e-06	11.08
kodim03	51.295	7.421e-06	9.18
kodim04	50.381	9.161e-06	8.18
kodim05	45.464	2.842e-05	7.57
kodim06	47.576	1.747e-05	7.63
kodim07	51.284	7.441e-06	7.25
kodim08	44.331	3.689e-05	4.72
kodim09	49.925	1.017e-05	7.49
kodim10	49.187	1.206e-05	4.78
kodim11	49.219	1.197e-05	4.76
kodim12	51.296	7.420e-06	5.09
kodim13	43.686	4.279e-05	4.70
kodim14	47.422	1.811e-05	4.86
kodim15	50.013	9.970e-06	4.79
kodim16	50.437	9.043e-06	4.73
kodim17	49.733	1.063e-05	4.66
kodim18	45.706	2.688e-05	4.66
kodim19	48.109	1.546e-05	4.69
kodim20	48.614	1.376e-05	4.67
kodim21	48.056	1.565e-05	4.70
kodim22	47.083	1.957e-05	4.83
kodim23	51.457	7.149e-06	4.75
kodim24	44.213	3.791e-05	4.72
Mean	48.407	–	6.11
Median	48.901	–	4.78
Std. dev.	2.391	–	–

A.5.2 Evaluation protocol

All 24 Kodak images are represented independently. RGB intensities are normalized to $[0, 1]$ and the corresponding two-dimensional coordinates are mapped to $[-1, 1]^2$. The reconstruction quality is measured over the full image by,

$$\mathrm{PSNR} = -10 \log_{10} (\mathrm{MSE}) , \tag{94}$$

Each branch is trained using Adam for 500 joint-optimization epochs independently, with an initial learning rate of 10^{-4}, a coordinate batch size of 8192, and cosine annealing to 10^{-6}. Three sinusoidal hidden layers of a width of 256 are present in each branch. $m = 6$ is taken as the initial high-frequency multiplier. Horizontal and vertical window lengths of 64 are used for SSA windows, retaining 16 horizontal and 8 vertical eigentriples for corresponding low-frequency reconstructions. The SAA weight initially decayed from 0.03 to 0.003, The weighted contribution of the SSA weight is capped at 25% of the reconstruction loss. The resulting model contains 533,528 trainable parameters.

Every 25 epochs, the full image is evaluated. The checkpoint with the highest full-image PSNR is saved and used for reporting. This implementation loads the best checkpoint reconstruction before the final evaluation. Therefore, the `final_psnr` and `best_psnr` columns are identical for all images in the supplied CSV file. These two values should not be considered as independent measurements. As such, Table 1 shows only the best checkpoint PSNR.

A.5.3 PRE-IMAGE RECONSTRUCTION RESULTS

Considering outputs of all the Kodak images from the proposed method, the mean of the PSNR distribution is $48.407\,\mathrm{dB}$. The distribution had a median of $48.901\,\mathrm{dB}$, and a standard deviation of was on $2.391\,\mathrm{dB}$. The lowest result was obtained on `kodim13` of $43.686\,\mathrm{dB}$, while the highest was on `kodim23` of $51.457\,\mathrm{dB}$. The total recorded run time for all 24 images was approximately 146.6 minutes, giving a mean run time of 6.11 minutes per image, with a median of 4.78 minutes.

A.5.4 COMPARISON PROTOCOL

The publicly available implementations released by the authors of COSMO-RC, INCODE, FR-INR, and SIREN were used to obtain external baseline results. Our local servers were used to execute every method and evaluate them on the Kodak dataset using full-image PSNR with RGB values normalized to [0, 1].

The same four-branch network architecture, hidden width, network depth, sigmoid-gated additive fusion, optimiser, joint-training budget, and checkpoint-selection procedure are used by the architecturally matched unguided baseline. It contains fixed role-aligned coordinate multipliers but lacks SSA-branch supervision, SSA-derived gate-prior supervision, branch specialization pretraining, PSO initialization, and scale learning

The complete proposed training and specialization strategy against an unguided parallel-branch baseline is evaluated through this comparison. The combined effect of SSA-derived branch supervision, gate-prior guidance,branch-specialized pretraining, and the proposed coordinate-scale optimization procedure is reflected by the observed difference.

A.5.5 MACHINE-READABLE SUPPLEMENTARY FILE

A machine-readable supplement to Table 1 is included, named `kodak_summary.csv`. This contains the image name, best-checkpoint PSNR and MSE, runtime, training hyperparameters, and total parameter count recorded by the implementation.

A.6 ONE-DIMENSIONAL AUDIO REPRESENTATION

This method was applied to one-dimensional audio signals with a reduced branch set $\mathcal{B}_{1D} = \{S, D\}$ to explore whether the approach extends beyond images. Figure 6 shows how the reconstruction matches the original waveform and preserves the dominant spectral content of the original.$c_S(t)$ captures the smoother temporal structure, whereas $c_D(t)$ represents rapid variation and higher-frequency details, which complements the temporal features. This shows that SSA-guided branch specialization can be applied not only to 2-dimensional images but also to one-dimensional continuous signals.

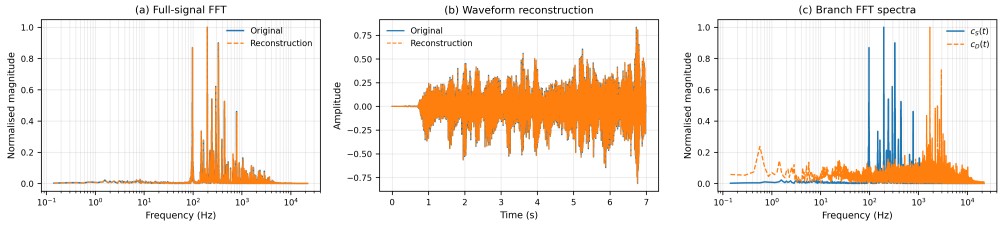


Figure 6: Audio signal representation with SSA-guided branch specialisation. (a) FFT magnitude comparison between the original and reconstructed waveform. (b) Time-domain waveform reconstruction. (c) FFT spectra of the learned gated branch contributions $c_S(t)$ and $c_D(t)$, corresponding to slow temporal structure and temporal detail.

The one-dimensional extension used for audio experiments is described in this appendix. The formulation is reduced from 4 directional roles to 2 temporal roles, where the low-rank temporal components are modeled by one branch while the other branch models the rest. The reconstructed waveform is obtained by adding up the two gated branches.

A.6.1 AUDIO PREPROCESSING

The input waveform is loaded from a WAV file and converted to a single-channel floating-point signal. This is supported for PCM WAV files with 8-, 16-, 24-, or 32-bit integer samples and 32- or 64-bit floating-point WAV files. If the signal is in multiple channels they are averaged out to get a mono waveform. The resulting samples are clipped to $[-1, 1]$ while the native sampling rate is preserved. Unless specified, the entire waveform is used. Let

$$s = [s_1, \dots, s_T]^\mathsf{T} \in [-1, 1]^T \tag{95}$$

denote the resulting waveform with T samples. Sample locations are mapped uniformly to

$$t_i \in [-1, 1], \qquad i = 1, \dots, T, \tag{96}$$

and the INR learns the continuous mapping

$$\widehat{s}_\theta : [-1, 1] \to \mathbb{R}. \tag{97}$$

The fixed random seed 42 is used for all stochastic operations, including network initialization, minibatch shuffling, randomized SSA, and optional PSO initialization,

A.6.2 ONE-DIMENSIONAL SSA-TARGETS

A one-dimensional SSA is used to decompose the waveform into low-rank structural components and a complementary detail component.

$$s = s_{\mathrm{S}}^{\mathrm{SSA}} + s_{\mathrm{D}}^{\mathrm{SSA}}. \tag{98}$$

A trajectory matrix is formed using a window length of

$$L = 256, \tag{99}$$

and the first

$$r = 16 \tag{100}$$

eigentriples are retained for the structural target. The detail target is defined as the exact residual:

$$s_{\mathrm{S}}^{\mathrm{SSA}} = \mathcal{S}_{\mathrm{L}}[s], \tag{101}$$

$$s_{\mathrm{D}}^{\mathrm{SSA}} = s - s_{\mathrm{S}}^{\mathrm{SSA}}. \tag{102}$$

Randomized SVD with eight oversampling vectors and one power iteration is used to compute the retained subspace for long audio signals.The diagonal averaging is used to map the reconstructed low-rank trajectory matrix to the waveform domain. The two audio roleS are therefore

$$\mathcal{B}_{1\mathrm{D}} = \{\mathrm{S}, \mathrm{D}\}, \tag{103}$$

where S denotes the dominant smooth temporal structure and D denotes the remaining rapid temporal variation and higher-frequency detail.

A.6.3 TWO-BRANCH AUDIO INR

Each role is represented by a SIREN branch

$$F_b : \mathbb{R} \to \mathbb{R}^2, \qquad b \in \{\mathrm{S}, \mathrm{D}\}. \tag{104}$$

A scaled temporal coordinate is received by the branch

$$\widetilde{t}_b = \rho_b t, \tag{105}$$

where the positive scale is parameterised as

$$\rho_b = \exp(\eta_b). \tag{106}$$

and the structural and detail scales are initialized to

$$\rho_{\mathrm{S}}^{(0)} = 100, \qquad \rho_{\mathrm{D}}^{(0)} = 400, \tag{107}$$

respectively. Each branch consists of 3 sinusoidal hidden layers of width 256. The first sinusoidal layer uses $\omega_0 = 60$, and the following layers use $\omega_0 = 30$. The raw waveform and the gate logit are given as the output.

$$F_b(\widetilde{t_b}) = [y_b(t), \alpha_b(t)]. \tag{108}$$

The independent gate is

$$g_b(t) = g_{\min} + (1 - g_{\min})\sigma\left(\frac{\alpha_b(t)}{\tau}\right), \tag{109}$$

with gate temperature

$$\tau = 1 \tag{110}$$

and gate floor

$$g_{\min} = 0.02. \tag{111}$$

The actual contribution of branch b is

$$c_b(t) = g_b(t)y_b(t), \tag{112}$$

and the final waveform is reconstructed by additive fusion:

$$\widehat{s}(t) = c_{\mathrm{S}}(t) + c_{\mathrm{D}}(t). \tag{113}$$

A.6.4 SSA-DERIVED TEMPORAL GATE PRIORS

The local magnitudes of two SSA targets are used to derive the gate priors.

$$A_b(t) = \left|s_b^{\mathrm{SSA}}(t)\right|. \tag{114}$$

The corresponding prior is

$$G_b^{\mathrm{SSA}}(t) = \left[\mathrm{clip}\left(\frac{A_b(t)}{\max_{j\in\{\mathrm{S,D}\}} A_j(t) + \epsilon}, 0, 1\right)\right]^{\gamma}, \tag{115}$$

where

$$\epsilon = 10^{-8}, \qquad \gamma = 0.75. \tag{116}$$

For any sample where the magnitude of both SSA components is below ϵ, the structural prior will be set to one and the detail prior will be zero. As in the image formulation, the two priors are independent and are not constrained to sum to one

A.6.5 BRANCH-SPECIALIZED PRETRAINING

The two branches are trained for 500 epochs independently. The pretraining objective for branch b is,

$$\mathcal{L}_{\mathrm{spec},b} = frac1T \sum_{i=1}^{T} \left[\left\|c_b(t_i) - s_b^{\mathrm{SSA}}(t_i)\right\|_2^2 + \lambda_{\mathrm{pg}} \left|g_b(t_i) - G_b^{\mathrm{SSA}}(t_i)\right|^2 \right]. \tag{117}$$

with

$$\lambda_{\mathrm{pg}} = 5 \times 10^{-4}. \tag{118}$$

Adam is used to optimize each branch separately with a learning rate of 10^{-4}. This stage does not use the fused waveform reconstruction loss. this establishes the initial structural/detail division of responsibility.

A.6.6 JOINT OPTIMIZATION

After branch-specialized pretraining, both branches are jointly optimised.The full-signal reconstruction loss is

$$\mathcal{L}_{\mathrm{rec}} = \frac{1}{T}\sum_{i=1}^{T} \left|\widehat{s}(t_i) - s_i\right|^2. \tag{119}$$

The branch-specialization loss is

$$\mathcal{L}_{\mathrm{SSA}} = \sum_{b\in\{\mathrm{S,D}\}} \frac{1}{T}\sum_{i=1}^{T} \left|c_b(t_i) - s_b^{\mathrm{SSA}}(t_i)\right|^2. \tag{120}$$

The joint gate-prior loss is

$$\mathcal{L}_{\text{gate}} = \frac{1}{2T} \sum_{i=1}^{T} \sum_{b \in \{\text{S,D}\}} \left| g_b(t_i) - G_b^{\text{SSA}}(t_i) \right|^2, \tag{121}$$

and the raw-component amplitude penalty is

$$\mathcal{L}_{\text{amp}} = \frac{1}{2} \sum_{b \in \{\text{S,D}\}} \frac{1}{T} \sum_{i=1}^{T} |y_b(t_i)|^2. \tag{122}$$

The full objective is

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \widetilde{\lambda}_{\text{SSA}}(k)\mathcal{L}_{\text{SSA}} + \lambda_g(k)\mathcal{L}_{\text{gate}} + \lambda_a \mathcal{L}_{\text{amp}}, \tag{123}$$

where k denotes the joint-training epoch. The SSA weight is linearly decayed from

$$0.03 \quad \text{to} \quad 0.003, \tag{124}$$

where the weighted contribution is adaptively capped at 25% of the current reconstruction loss. The gate prior weight is linearly decayed from

$$2 \times 10^{-4} \quad \text{to} \quad 0, \tag{125}$$

and the raw-component amplitude penalty is weighted by

$$\lambda_a = 10^{-8}. \tag{126}$$

No additional orthogonality penalty is used in the reported configuration.

Adam is used to perform joint optimization for 1000 epochs with a learning rate of 9×10^{-5}, cosine annealing to 10^{-6}, and with a coordinate batch size of 8192. Every 25 epochs, the complete waveform is evaluated, and the checkpoint with the lowest full signal MSE is kept for final evaluation.

A.6.7 PSO-BASED SCALE INITIALIZATION

A PSO-based initialization of the two coordinate scales was used for the reported experiments.

$$\boldsymbol{q} = [\rho_{\text{S}}, \rho_{\text{D}}]. \tag{127}$$

The default role-specific search intervals are

$$\rho_{\text{S}} \in [0.5, 500], \qquad \rho_{\text{D}} \in [1, 2000]. \tag{128}$$

12 particles and 6 iterations are used for the search. Each particle is evaluated by 25 epochs of short SSA-guided fitting using 25% of the samples for fitting and 25% for fitness evaluation. The PSO inertia, cognitive, and social coefficients are 0.6, 1.5, and 1.5, respectively. The selected scales are frozen during branch pretraining and the early part of joint optimization when PSO initialization is used and are only unfrozen after 70% of the joint training epochs.

A.6.8 EVALUATION AND DIAGNOSTIC ANALYSIS

The accuracy of the waveform is measured using full-signal MSE and

$$\text{PSNR} = -10 \log_{10}(\text{MSE}), \tag{129}$$

which follows the implementation for the normalized waveform. The full waveform MSE is considered when selecting the best checkpoint rather than minibatch loss.

One-sided real FFT is used for frequency domain comparisons.The magnitude is normalized by the number of waveform samples, doubling the non-DC and non-Nyquist bins, and is displayed as

$$20 \log_{10} \left(\max(|S(f)|, 10^{-12}) \right). \tag{130}$$

The original and reconstructed waveforms and spectra, the SSA targets and learned gated contributions, the learned and target gate signals, and the evolution of the branch coordinate scales are compared in the analysis.The slow-varying dominant temporal structure is captured by. The structural contribution $c_{\text{S}}(t)$ while the detail contribution cD(t) captures the complementary rapid variation.Their sum gives the reconstructed waveform inequation 113. The experiment is intended as a proof of concept that explicit branch specialisation extends beyond two-dimensional images. It evaluates continuous audio representation rather than perceptual coding, compression, denoising, or source separation

Table 4: Computational profile of the SSA-guided INR on `kodim20.png`. Forward GFLOPs count basic arithmetic for one full-image model pass.

Method	Params (K)	Fwd. GFLOPs	Joint train (s/epoch)	Inference (s/image)	Throughput (GFLOPs/s)	PSNR (dB)
SSA-guided INR	533.528	418.373	0.1896	0.0425	9837.38	48.10

PSO initialisation took 72.342 s, branch pretraining 89.914 s, and joint training including evaluations 95.940 s. The full run before the inference benchmark took 262.022 s. The measured SSA target stage took 0.135 s; its cache status should be stated. Forward GFLOPs exclude SVD and nonlinear-function costs. Throughput is forward GFLOPs divided by the unrounded inference time.

Method	Decomposition setting	Params (K)	PSNR (dB)	ΔPSNR (dB)	Runtime (s)
SSA	Windows $(64, 64)$; ranks $(16, 8)$	533.528	48.61	+0.15	262.0
FFT	Nyquist cutoffs $(0.25, 0.25)$	533.528	48.29	+0.19	593.3
DWT	One-level Haar, periodization	533.528	48.31	+0.21	261.2
KLT	Centered ranks $(16, 8)$	533.528	47.45	-0.65	260.0

Table 5: Decomposition comparison on `kodim20.png`. PSNR differences use the reported SSA value as the reference. Runtime is the reported duration of each run, rather than the cost of the decomposition alone.

A.6.9 AUDIO SUPPLEMENTARY MEDIA

We have provided the original audio, the reconstructed audio, and individual gate branch contributions as supplementary media files to complement the waveform and frequency-domain visualization.The structural-branch file contains $c_{\mathrm{S}}(t) = g_{\mathrm{S}}(t)y_{\mathrm{S}}(t)$, while the detail-branch file contains $c_{\mathrm{D}}(t) = g_{\mathrm{D}}(t)y_{\mathrm{D}}(t)$. These are the actual branch contributions used in the additive reconstruction,

$$\widehat{s}(t) = c_{\mathrm{S}}(t) + c_{\mathrm{D}}(t). \tag{131}$$

The dominant smooth temporal content is preserved by the structural contribution, while the complementary rapid variation and higher-frequency structure are contained in the detail contribution. The media files are provided as mono WAV signals at the sampling rate of the original input signal. These files are to support qualitative investigation of the learned division of responsibility between the two specialized branches

- `audio_original.wav`: original input waveform;
- `audio_reconstruction.wav`: final additive reconstruction;
- `audio_structural_branch.wav`: gated structural contribution $c_{\mathrm{S}}(t)$;
- `audio_detail_branch.wav`: gated detail contribution $c_{\mathrm{D}}(t)$.

A.7 COMPUTATIONAL PROFILE

Computational profile of the proposed method is given in Table 4.

A.8 ALTERNATIVE SIGNAL DECOMPOSITIONS

We replaced SSA with a two-dimensional FFT, a one-level Haar DWT, or a centered KLT while retaining the four-branch model and training procedure. All runs used `kodim20.png` (768×512), 533,528 trainable parameters, 500 branch-pretraining epochs, 500 joint-training epochs, and PSO initialization. Table 5 reports the best full-image checkpoint from each run.

On this image, FFT and DWT achieved approximately 0.2 dB higher PSNR than SSA, whereas KLT at the selected ranks achieved 0.65 dB lower PSNR. The FFT and DWT results differ by only 0.0153 dB. Their additive reconstruction errors, like that of KLT, were at most 5.96×10^{-8} in maximum absolute value. These results show that the branch targets can be formed with other decompositions, but a single-image result does not establish a general ranking. The observed runtime

difference should likewise not be attributed solely to the transform without matched profiling of the decomposition and training stages.