STABLE INITIALIZATION WITHOUT THE CLT

Anonymous authors
Paper under double-blind review

ABSTRACT

Successful training of deep neural networks is highly dependent on the distribution of the initial weights. If the weights are too large, network training blows up; if they are too small, the model fails to learn features. Stable initialization is the optimal moderation between these two extremes. The conventional theory of random networks uses the Central Limit Theorem to control inter-neuron dependencies, which introduces distributional approximation error and coupling between layers. For networks with sine activations, we derive the uniform-phase initialization, which obviates distributional approximation and fully decouples the layers. Ours is the first work to use the sine function's periodic symmetry. Models trained with the uniform-phase initialization outperform the state of the art in neural representation tasks like image and audio fitting. We find that our untuned models are competitive with the best-tuned baselines from previous work and support μP width scaling.

1 INTRODUCTION

Today it is widely accepted that increasing a model's representational capacity by increasing its width N and depth L leads to stronger performance on complex learning tasks (Hestness et al., 2017). Two decades ago, however, training large models was a serious concern: wide and deep networks are highly sensitive to the initialization of their parameters (Hastie et al., 2009, §11.5.1) and suffer from exploding or vanishing gradients, which make first-order optimization challenging (Glorot & Bengio, 2010). These so-called gradient instabilities motivated the search for **stable initialization** distributions for the weights.

We study a **multilayer perceptron** $f : \mathbb{R}^{N_{\text{in}}} \to \mathbb{R}^{N_{\text{out}}}$, which is the composition $f(x) = (g_{\text{out}} \circ h \circ g_{\text{in}})(x)$, comprising

$$\begin{aligned}
g_{\text{in}} &: \mathbb{R}^{N_{\text{in}}} \to \mathbb{R}^{N}, & x &\mapsto \sigma(W_{\text{in}}x + b_{\text{in}}), && \text{the } \textbf{input layer}; \\
g_{\text{out}} &: \mathbb{R}^{N} \to \mathbb{R}^{N_{\text{out}}}, & x &\mapsto W_{\text{out}}x + b_{\text{out}}, && \text{the } \textbf{output layer}; \text{ and} \\
h &: \mathbb{R}^{N} \to \mathbb{R}^{N}, & x &\mapsto (h_L \circ h_{L-1} \circ \cdots \circ h_1)(x), && \text{the } \textbf{hidden layers}, \text{ defined by} \\
h_\ell &: \mathbb{R}^{N} \to \mathbb{R}^{N}, & x &\mapsto \sigma(W_\ell x + b_\ell), && \text{for } \ell \in \{1, 2, \ldots, L\}.
\end{aligned}$$

Stable initialization encompasses two questions: (1) how to tailor the hidden layers to the activation function and (2) how to tailor the input and output layers to the task.

Hidden layers Early theory required that as $L, N \to \infty$, the initial network's regularity properties scale as $O(1)$, i.e. not "blow up" (LeCun et al., 2012; Glorot & Bengio, 2010; He et al., 2015). The edge-of-chaos school argues that "blowdown" is equally pathological and strengthened this desideratum to $\Theta(1)$ (Poole et al., 2016; Schoenholz et al., 2017; Yang & Schoenholz, 2017; Hayou et al., 2019). To understand why this requirement is generally nontrivial for hidden layers, we express the Jacobian of h at a given input:

$$\frac{\partial}{\partial x} h(x) = \operatorname{diag}(\sigma'(z_L))W_L \cdots \operatorname{diag}(\sigma'(z_1))W_1, \tag{1}$$

where z_ℓ is the preactivation at hidden layer ℓ. The activation slopes $\operatorname{diag}(\sigma'(z_\ell))$, $\ell \in \{1, \ldots, L\}$, depend on the preactivation distribution, which in turn depends on the previous layer's input and weights.

Managing these dependencies demands sophisticated analysis, usually in four steps:

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.

1. Identify the criteria to stabilize, usually preactivation scales and gradient norms (LeCun et al., 2012; Glorot & Bengio, 2010; Saxe et al., 2014; He et al., 2015; Hanin & Rolnick, 2018; Pennington et al., 2017; Pennington & Worah, 2018; Xiao et al., 2018; Hayou et al., 2021; Yang et al., 2022).
2. Calculate the criteria, approximately or exactly, for a location-scale parametric family, usually Normal, of preactivation distributions (Poole et al., 2016; Schoenholz et al., 2017; Klambauer et al., 2017; Hayou et al., 2019). This step usually involves calculating or approximating $\mathbb{E}\,\sigma(Z)$, $\mathrm{Var}\sigma(Z)$, $\mathbb{E}\,\sigma'(Z)$, and $\mathrm{Var}\sigma'(Z)$, where Z is a random variable that approximates the preactivation.
3. Choose distributions for W_ℓ and b_ℓ such that the preactivation distributions satisfy a layerwise fixed-point recursion, provided that they are Normally distributed (LeCun et al., 2012; Glorot & Bengio, 2010; He et al., 2015; Mishkin & Matas, 2016; Hayou et al., 2021).
4. Apply the Central Limit Theorem to argue that in the large-N limit, the preactivations are approximately Normal (Poole et al., 2016; Schoenholz et al., 2017; Yang & Schoenholz, 2017; Lee et al., 2017; Jacot et al., 2018; Yang, 2019). This closes the layerwise recursion.

The Central Limit Theorem is the weakest link in this theory. It incurs an error of $O(N^{-1})$, and narrow networks are appreciably non-Gaussian at initialization (Wolinski & Arbel, 2022). One solution, pursued in the monograph by Roberts et al. (2022), is to expand to second order, reducing the asymptotic error to $O(N^{-2})$.

We focus on the activation function $\sin(x)$. The conventional route is taken by Combette et al. (2026), who instantiate the fixed-point recursion (Step 3) as a transcendental equation involving the Lambert W function. The second-order theory of Roberts et al. (2022, §5.3.3,§9.3) classifies $\sin$ in the same universality class as $\tanh$. Altogether, previous work on stable-initialization theory for $\sin(x)$ uses general properties of $\sin(x)$, such as its moments, Taylor series, and range, but—crucially—has ignored its periodicity.

Input and output layers The first and last layers of a neural network deserve their own attention. Whereas stable initialization theory is interested in the asymptotic *slope* (namely, zero, for edge-of-chaos initialization) of the network's behavior with respect to L and N, the absolute scale of the input and output weights and biases determines the *intercept*: no matter how large L and N are, $f(10x)$ is a very different function than $f(x)$. Sinusoidal neural networks are known to exhibit a fragile dependence on the weight scale in general (Yeom et al., 2025) and input weights in particular (Sitzmann et al., 2020; Combette et al., 2026). Many works tune a hyperparameter ω_0 that scales the inputs by an isotropic base frequency. (The Nyquist frequency can be motivated by sensor resolution (Yuce et al., 2022), but it follows the sensor rather than the underlying field.) Alsakabi et al. (2026) use a root-finding algorithm to match the initial network's spectrum to the discrete sine transform of the field, which is optimal in a sense, but requires rich information about the training data. Tancik et al. (2021) target the input frequency of a Fourier feature network as a meta-learning task.

1.1 Summary of our method

Conventional stable initialization theory models preactivations as distributed on $\mathbb{R}^N$ and centered at 0, which necessitates small or zero biases. However, because sine is a periodic function, the preactivation may equivalently be viewed as wrapped on the torus $[0, 2\pi]^N$, which has a unique translation-invariant probability measure. By sampling large biases, we spread the preactivations uniformly around the torus *irrespective of the location and scale of the previous layer*, which corresponds to drawing biases from $\mathcal{N}(0, \infty)$ in a conventional stable initialization (Appendix A).

Definition 1. *The* uniform-phase *initialization for hidden layers $x \mapsto \sin(Wx + b)$ is*

$$W_{ij} \sim \mathcal{N}(0, 2/N) \qquad \textit{independently, and}$$
$$b_i \sim \mathcal{U}([0, 2\pi]) \qquad \textit{independently.}$$

As a consequence, miraculously, all of the entries of all of the matrices in equation 1 become not only jointly independent but also independent of the input (Lemma 1). The essence is summarized by the following fact:

> Let X have *any* distribution on $\mathbb{R}$, and let $b \sim \mathcal{U}([0, 2\pi])$ be independent of X.
> Then $\cos(X + b)$ is independent of X.

We state and prove the neural network version (Lemma 1) and then deploy it to prove our main result (Theorem 1) on the stability of hidden layers. Our hidden-layer theory facilitates an auxiliary result on initializing the input and output layers (Section 3), which lends interpretability to the input and output scales by relating them to non-asymptotic average-case regularity properties of the initial network.

Altogether, for every fixed input x, our initialization distribution guarantees:

$$\mathbb{E}\, J^\intercal J = \mathbb{E}\, J J^\intercal = I,$$

simultaneously for all Jacobians J between *any* two hidden layers, and

$$\mathbb{E}\, f(x), \quad \mathrm{Cov} f(x), \quad \text{and} \quad \mathbb{E}\, \nabla f(x) \left(\mathrm{Cov} f(x)\right)^{-1} (\nabla f(x))^\intercal$$

all attain prescribed values exactly. They can be imposed as an inductive bias or moment-matched to data.

1.2 Contributions

We propose a simple and—to our knowledge, without precedent—exact solution to stable initialization at any width and depth that harnesses the periodicity of sinusoidal activation. We also propose an exact moment-matching criterion for zero-shot tuning of the input and output layer initialization. Our method is architecture-agnostic and adapts to the spatial resolution and output statistics of the neural representation task's target. Untuned, it outperforms fine-tuned baselines for sinusoidal network initialization.

We validate the conventional wisdom on the effectiveness of stable initialization by comparing our initialization to stable initializations in the literature. On most examples, our method meets or exceeds the bar set by previous work. We also test the learning-rate scaling of the maximal-update parametrization (μP) for neural representations and question the necessity of its architecture-dependent last-layer scaling.

1.3 Related work

Sinusoidal activations and neural representation The sine function satisfies the hypotheses of universal approximation theory in neural networks; in fact, similar guarantees can be obtained using the older and more robust theory of Fourier analysis (Parascandolo et al., 2017). However, multilayer networks with sinusoidal activation have proved remarkably effective for the task of **neural representation**: learning real-world signals represented as functions of an input coordinate, e.g. audio as a map from time to sound pressure, or an image as a map from spatial coordinates to pixel values (Sitzmann et al., 2020; Dupont et al., 2021; Strümpler et al., 2022). A related idea (Fourier features) is to use sinusoids in the input layer, but with fixed isotropic weights (Tancik et al., 2020; Mildenhall et al., 2020).

Initializing sinusoidal networks Sitzmann et al. (2020) (henceforth SM20) attempt to impose zero mean and unit variance on the preactivation distribution but do not account for gradient blowup with network depth. Although input-frequency tuning partly mitigates this weakness from an NTK perspective (Belbute-Peres & Kolter, 2023), Combette et al. (2026) (henceforth CVP26) are the first to carry out the complete stable-initialization procedure to control both preactivations and gradients. Novello et al. (2025); Yeom et al. (2025) are explicitly interested in controlling the frequency power spectrum of the initialized network. Alsakabi et al. (2026) deterministically match it to the training data. Ben-Shabat et al. (2022) propose an initialization geared specifically toward signed distance functions.

Scaling, feature learning, and μP Maximal update parametrization (μP) is a theory for scaling hyperparameters with width N at a fixed depth (the main use case is to tune a model's hyperparameters at a small scale and subsequently transfer them to a much wider model) (Yang et al., 2022; 2024; Dey et al., 2026). But it only gives the slope and not the intercept; for sine-activated networks, the

absolute scales depend on the task and have an outsized effect on training (Yeom et al., 2025). While the predictions of μP are asymptotic $\Theta(\cdot)$- and $O(\cdot)$-type results, our results are exact equalities for any width and depth. μP's learning-rate scaling is orthogonal to our work, but we test it nevertheless on our initializations. μP's initialization scaling agrees with our exact parameters in all but the final layer; we discuss this subtlety in Section 4.5.

2 Theory of the uniform-phase initialization for hidden layers

Specializing the hidden map h from Section 1 to $\sigma = \sin$, let $x_0 = x$ and write the layerwise pre- and post-activations as

$$z_\ell = W_\ell x_{\ell-1} + b_\ell, \qquad x_\ell = \sin(z_\ell), \qquad \ell \in [1 \dots L], \tag{2}$$

so that $h(x) = x_L$. Introduce the notation $\operatorname{Cos} x = \operatorname{diag}(\cos(x))$, where $\cos$ is applied elementwise. All expectations in this section are over the initialization distribution, with x_0 arbitrary.

We now state and prove a few useful properties of the uniform-phase initialization.

Lemma 1 (Independent Jacobian factorization)**.** *In the uniform-phase initialization, the terms* $\{W_\ell\}_{\ell=1}^L \cup \{\cos z_\ell\}_{\ell=1}^L$ *are independent.*

Proof. Let $\phi_1, \dots, \phi_L : \mathbb{R}^N \to \mathbb{R}$ and $\psi_1, \dots, \psi_L : \mathbb{R}^{N\times N} \to \mathbb{R}$ be arbitrary indicator functions of measurable sets.

$$\begin{aligned}
&\mathbb{E} \prod_{\ell=1}^{L} \phi_\ell(\cos(z_\ell))\psi_\ell(W_\ell) \\
&\quad= \mathbb{E}\, \phi_L(\cos(W_L x_{L-1} + b_L))\psi_L(W_L) \prod_{\ell=1}^{L-1} \phi_\ell(\cos(W_\ell x_{\ell-1} + b_\ell))\psi_\ell(W_\ell) \\
&\quad= \mathbb{E} \underbrace{\mathbb{E}_{b_L}\left[\phi_L(\cos(W_L x_{L-1} + b_L))\right]}_{=\text{constant}} \psi_L(W_L) \prod_{\ell=1}^{L-1} \phi_\ell(\cos(W_\ell x_{\ell-1} + b_\ell))\psi_\ell(W_\ell) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(conditioning on all randomness other than } b_L\text{)} \\
&\quad= \mathbb{E}\left[\phi_L(\cos(W_L x_{L-1} + b_L))\right] \mathbb{E}\, \psi_L(W_L) \prod_{\ell=1}^{L-1} \phi_\ell(\cos(W_\ell x_{\ell-1} + b_\ell))\psi_\ell(W_\ell) \qquad (\star) \\
&\quad= \mathbb{E}\left[\phi_L(\cos(W_L x_{L-1} + b_L))\right] \mathbb{E}\, \psi_L(W_L)\, \mathbb{E} \prod_{\ell=1}^{L-1} \phi_\ell(\cos(W_\ell x_{\ell-1} + b_\ell))\psi_\ell(W_\ell) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (W_L \text{ is independent}) \\
&\quad= \prod_{\ell=1}^{L} \mathbb{E}\, \phi_\ell(\cos(z_\ell))\, \mathbb{E}\, \psi_\ell(W_\ell) \qquad\qquad\qquad\qquad\qquad\qquad (\text{induction on } L)
\end{aligned}$$

The crucial step, $(\star)$, uses the fact that $W_L x_{L-1} + b_L$ is uniformly distributed modulo 2π for any fixed $W_L x_{L-1}$. Averaging over b_L therefore yields a constant, which can be pulled out of the expectation. □

Lemma 2 (Moments of uniform-phase sinusoids)**.** *Let* b *be uniformly distributed on the torus* $[0, 2\pi]^N$*. Then* $\mathbb{E}\cos(b) = 0$ *and* $\mathbb{E}\left[\cos(b)\cos(b)^\intercal\right] = \frac{1}{2}I$*.*

Proof. By circular symmetry in each coordinate, $\mathbb{E}\cos(b) = 0$. Because the coordinates are independent, the off-diagonal entries of $\mathbb{E}\left[\cos(b)\cos(b)^\intercal\right]$ are zero. On the diagonal, $\mathbb{E}\cos^2 b_1 = \mathbb{E}\sin^2 b_1$ by circular symmetry, while $\mathbb{E}\cos^2 b_1 + \mathbb{E}\sin^2 b_1 = 1$ by the Pythagorean identity; therefore $\mathbb{E}\cos^2 b_1 = \frac{1}{2}$. □

Combining these inductively proves our main result on hidden layers.

Theorem 1. *Let $0 \le s \le t \le L$ be integers. Then for all inputs x_0, the hidden layers equation 2 satisfy*

$$\mathbb{E}\left(\frac{\partial x_t}{\partial x_s}\right)\left(\frac{\partial x_t}{\partial x_s}\right)^\intercal = I, \qquad \mathbb{E}\left(\frac{\partial x_t}{\partial x_s}\right)^\intercal\left(\frac{\partial x_t}{\partial x_s}\right) = I. \tag{3}$$

Write $\delta_s^t \coloneqq \dfrac{\partial x_t}{\partial x_s}$. By the chain rule, δ_s^t satisfies both a forward recursion in t (with s fixed) and a backward recursion in s (with t fixed):

$$\delta_s^t = \operatorname{Cos} z_t W_t \delta_s^{t-1}, \qquad \delta_s^s = I, \tag{4a}$$

$$\delta_s^t = \delta_{s+1}^t \operatorname{Cos} z_{s+1} W_{s+1}, \qquad \delta_t^t = I. \tag{4b}$$

Note that δ_s^t depends only on x_s, $\{W_\ell\}_{\ell=s+1}^t$, and $\{b_\ell\}_{\ell=s+1}^t$; we shall use this fact to argue for independence in later steps.

Proof that $\mathbb{E}\,\delta_s^t(\delta_s^t)^\intercal = I$. We prove this by induction on t, with s fixed. The base case $t = s$ is immediate from equation 4a. For the inductive step, we assume that $\mathbb{E}\,\delta_s^{t-1}(\delta_s^{t-1})^\intercal = I$ and want to show that $\mathbb{E}\,\delta_s^t(\delta_s^t)^\intercal = I$. By equation 4a,

$$\delta_s^t(\delta_s^t)^\intercal = \operatorname{Cos} z_t W_t \delta_s^{t-1}(\delta_s^{t-1})^\intercal W_t^\intercal \operatorname{Cos} z_t. \tag{5}$$

According to Lemma 1, δ_s^{t-1}, W_t, and $\operatorname{Cos} z_t$ are mutually independent. Therefore, we may evaluate this expectation from the inside out by repeatedly conditioning on the outer terms:

$$\begin{aligned}
\mathbb{E}\,\delta_s^t(\delta_s^t)^\intercal &= \mathbb{E}\operatorname{Cos} z_t W_t \left(\mathbb{E}\,\delta_s^{t-1}(\delta_s^{t-1})^\intercal\right) W_t^\intercal \operatorname{Cos} z_t && (6)\\
&= \mathbb{E}\operatorname{Cos} z_t W_t W_t^\intercal \operatorname{Cos} z_t && (7)\\
&= \mathbb{E}\operatorname{Cos} z_t \left(\mathbb{E}\,W_t W_t^\intercal\right) \operatorname{Cos} z_t && (8)\\
&= 2\,\mathbb{E}\operatorname{Cos} z_t \operatorname{Cos} z_t && (9)\\
&= I && (10)
\end{aligned}$$

by Lemma 2. □

Proof that $\mathbb{E}(\delta_s^t)^\intercal\delta_s^t = I$. We prove this by downward induction on s, with t fixed. The base case $s = t$ is immediate from equation 4b. For the inductive step, we assume that $\mathbb{E}(\delta_{s+1}^t)^\intercal\delta_{s+1}^t = I$ and want to show that $\mathbb{E}(\delta_s^t)^\intercal\delta_s^t = I$. By equation 4b,

$$(\delta_s^t)^\intercal\delta_s^t = W_{s+1}^\intercal \operatorname{Cos} z_{s+1}(\delta_{s+1}^t)^\intercal\delta_{s+1}^t \operatorname{Cos} z_{s+1} W_{s+1}.$$

Using the same conditioning tactic as above,

$$\begin{aligned}
\mathbb{E}(\delta_s^t)^\intercal\delta_s^t &= \mathbb{E}\,W_{s+1}^\intercal \operatorname{Cos} z_{s+1}\left(\mathbb{E}(\delta_{s+1}^t)^\intercal\delta_{s+1}^t\right)\operatorname{Cos} z_{s+1} W_{s+1}\\
&= \mathbb{E}\,W_{s+1}^\intercal \operatorname{Cos} z_{s+1} \operatorname{Cos} z_{s+1} W_{s+1}\\
&= \mathbb{E}\,W_{s+1}^\intercal \left(\mathbb{E}\operatorname{Cos} z_{s+1} \operatorname{Cos} z_{s+1}\right) W_{s+1}\\
&= \frac{1}{2}\,\mathbb{E}\,W_{s+1}^\intercal W_{s+1} = I.
\end{aligned}$$

□

This theorem guarantees that the Jacobian (or gradient) between any two layers is isotropic, and its average squared singular value is 1. In the backpropagation direction, early training loss gradients neither grow nor decay on average as they propagate backward through the network. In the forward direction, the network, as a mathematical function that maps inputs to outputs, possesses zeroth-order (range) and first-order (slope) properties that are independent of width and depth.

3 MOMENT MATCHING FOR INPUT AND OUTPUT LAYERS

Thus far, we have dealt only with h, the network's hidden-layer map, which has N inputs, N outputs, and depth L. To complete the network $f = g_{\text{out}} \circ h \circ g_{\text{in}}$ from $\mathbb{R}^{N_{\text{in}}}$ to $\mathbb{R}^{N_{\text{out}}}$ as defined in the introduction, we must initialize the input layer g_{in} and the output layer g_{out}. We draw b_{in} from a uniform distribution on $[0, 2\pi]$, as we do for the hidden-layer biases. This establishes translation invariance and eliminates the need to center the data.

For target moments μ, $\Sigma \succ 0$, and $\Omega \succeq 0$, we initialize the remaining three parameters $W_{\text{in}} \in \mathbb{R}^{N \times N_{\text{in}}}$, $W_{\text{out}} \in \mathbb{R}^{N_{\text{out}} \times N}$, and $b_{\text{out}} \in \mathbb{R}^{N_{\text{out}}}$ by moment matching:

$$\mu = \mathbb{E}\, f(x) = \mathbb{E}\, b_{\text{out}}, \qquad \text{the output mean,} \tag{11a}$$

$$\Sigma = \operatorname{Cov} f(x) = \tfrac{1}{2}\, \mathbb{E}\, W_{\text{out}} W_{\text{out}}^{\intercal}, \qquad \text{the output covariance, and} \tag{11b}$$

$$\Omega = \mathbb{E}\, \frac{\partial f(x)}{\partial x}^{\intercal} \Sigma^{-1} \frac{\partial f(x)}{\partial x} = \frac{N_{\text{out}}}{N}\, \mathbb{E}\, W_{\text{in}}^{\intercal} W_{\text{in}}, \qquad \text{the normalized structure tensor.} \tag{11c}$$

Because the uniform-phase initialization is translation invariant, these moments may be interpreted as being evaluated at any particular x or over a representative distribution of x. After μ, Σ, and Ω are either prescribed or estimated from data, we match these target moments by setting

$$b_{\text{out}} = \hat{\mu}, \tag{12a}$$

$$W_{\text{out}} \sim \mathcal{N}(0, 2\hat{\Sigma}/N) \text{ columns,} \qquad \text{and} \tag{12b}$$

$$W_{\text{in}} \sim \mathcal{N}(0, \hat{\Omega}/N_{\text{out}}) \text{ rows.} \tag{12c}$$

The key idea is to use Ω to calibrate the input scale in the "physical" (that is, nonfrequency) domain. By initializing W_{out} and W_{in} in this way, we nondimensionalize both the inputs and the outputs for the hidden section of the neural network.

In prior work (Sitzmann et al., 2020; Combette et al., 2026), the input weights are scaled using a "characteristic frequency" ω_0: the Nyquist frequency (in audio), ~ 30 periods (in images), ~ 2 periods (in physics-informed neural networks). However, this and other scales have an outsized effect on learning and require tuning (Belbute-Peres & Kolter, 2023; Yeom et al., 2025). By contrast, matching the observed value of Ω allows the network to adapt to the regularity characteristics of the data.

4 NUMERICAL EXPERIMENTS

Stable initialization is a mathematical definition accompanied by a scientific hypothesis: a stably initialized network will perform better than an unstably initialized network after thousands of epochs under the same training operator. Our construction, which exactly calibrates initialization-ensemble second moments at any depth and width, establishes the ideal laboratory conditions in which to test this hypothesis. Appendix C gives full details of the tasks, preprocessing, model implementations, optimization, metrics, and hyperparameter sweeps.

4.1 IMAGE FITTING

We fit the Cameraman image on a 128×128 training grid and evaluate it on a 512×512 grid to assess generalization. We compare the following architectures (details in Appendix C):

SIREN (UniformPhase-MM) Our proposed initialization, which uses uniform-phase biases and moment-matched input and output layers, estimates μ, Σ, and Ω from the training data, and requires no hand-tuned ω_0.

SIREN (CVP26 $\sigma_a = 0$) The edge-of-chaos initialization of Combette et al. (2026).

SIREN (SM20) The original initialization of Sitzmann et al. (2020).

Tanh (PE-Xavier) A Tanh MLP with random Fourier-feature encoding (Tancik et al., 2020) and Xavier weights.

ReLU (Kaiming), SiLU (Xavier), GeLU (Xavier) Standard MLPs with the corresponding activation and initialization.

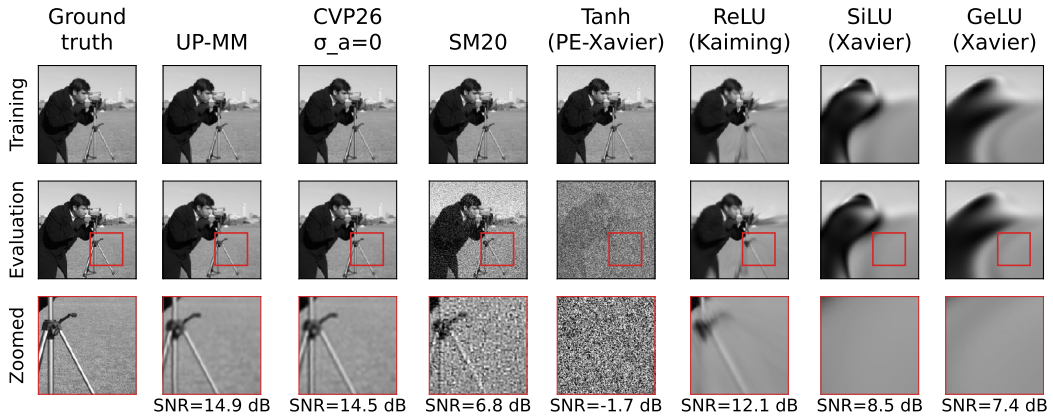

Figure 1: Image fitting on the Cameraman image. Columns: the ground truth followed by one column per initialization, with the evaluation SNR beneath each. Top row: 128×128 training image. Middle row: 512×512 test image. Bottom row: zoomed region (red square in the middle row).

The uniform-phase, moment-matched initialization achieves the lowest evaluation MSE by a small margin (Table 1 in Appendix D), followed by CVP26 ($\sigma_a = 0$) and the remaining replications of Combette et al. (2026). Visually, the top two methods both preserve fine details without introducing excessive noise (figure 1).

4.2 Audio fitting

We use the same architectures in a replication of the Bach audio experiment of Sitzmann et al. (2020, Appendix 8) [1]. For the CVP26 and SM20 SIRENs, we replicate the input frequency $\omega_0 = 30$ used by Sitzmann et al. (2020). For our SIREN, we use the moment-matched initialization, which estimates $\hat{\Omega}$ from the data and requires no hand-tuned frequency. The SM20 SIREN attains the highest fidelity, with an SNR of 31.15 dB, but our moment-matched SIREN reaches 29.33 dB *without* any hand-tuned frequency parameter; its input scale is automatically calibrated through $\hat{\Omega}$. Our SIREN's SNR exceeds CVP26's by nearly 20 dB, and the non-sinusoidal networks (ReLU, SiLU, GeLU) collapse to near-zero output (figure 5 in Appendix D).

4.3 Narrow networks

We repeat the experiment of §4.1 but reduce the width to $N = 16$ to stress the infinite-width assumption underlying the CVP26 and SM20 initializations. The differences between initializations are amplified as a result. Our moment-matched SIREN remains the best performer across all metrics (SNR: 10.28 dB), and the gap to CVP26 widens to 6.8 dB (figure 4 and Table 2 in Appendix D).

4.4 Moment matching (zero-shot) versus ω_0 tuning (hyperparameter search)

The hyperparameterizations of CVP26 and SM20 call for an isotropic "characteristic frequency" ω_0 to scale the inputs. Prior work suggests $\omega_0 = 64\pi/\sqrt{6}$ for a domain of $[-1, 1]$. By contrast, our method achieves zero-shot calibration to the anisotropic statistics of the image field; it is invariant to domain translation and equivariant under domain scaling, sampling resolution, and output scaling. We compare our method's zero-shot tuning to a detailed grid search for prior methods' ω_0.

[1] The signal is a 7 s mono audio signal at 44.1 kHz. The time coordinate is scaled to $[-100, 100]$, the audio is normalized by $\max |y|$, and the network is trained and evaluated on the *full* signal (no held-out test split). All networks have $L = 5$ hidden layers of width $N = 256$ and are trained for 5000 epochs with Adam at a learning rate of 5×10^{-5}. The Tanh network uses random Fourier features with $\sigma = 10$.

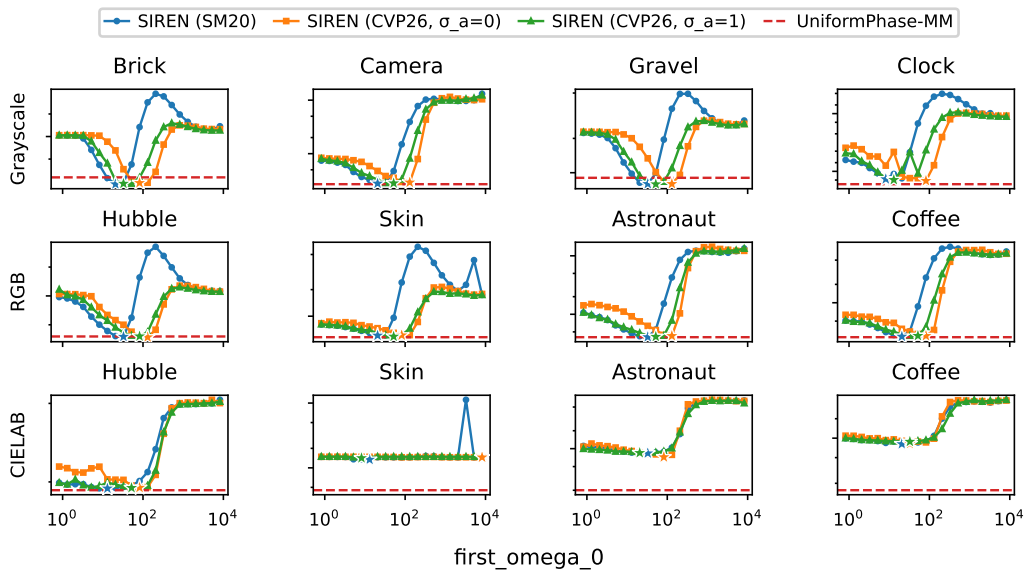

Figure 2: Grayscale and color ω_0 sweep. Test MSE versus input-layer frequency ω_0; rows are the grayscale, RGB, and CIELAB experiments, and columns are the four images in each. A star marks the minimum of every sweep, and the dashed red line marks the uniform-phase MM test MSE. Each panel has its own logarithmic vertical scale; for absolute errors, see Tables 3, 4, and 5.

Using the same architecture and training as §4.1, we evaluate on four **grayscale** images from `scikit-image` (Brick, Camera, Gravel, and Clock) and four color images (Hubble Deep Field, Skin, Astronaut, and Coffee), in both **RGB** and **CIELAB** color spaces. For each baseline SIREN initialization (SM20, CVP26 $\sigma_a = 0$, CVP26 $\sigma_a = 1$), we sweep the input-layer frequency ω_0 over four powers of 10 on a logarithmic grid of 21 values centered at prior work's ω_0. The estimated moment-matched parameters and per-image best test errors are reported in Appendix E, and the best-ω_0 images in the Supplementary Material, Appendix F.

Across the **grayscale** and **RGB** experiments, the moment-matched initialization achieves the lowest test MSE on 5 of 8 images (Camera, Clock, Skin, Astronaut, and Coffee). On the remaining 3 images (Brick, Gravel, and Hubble), it is competitive, losing to the best-tuned baseline by 19%, 12%, and 1.5%, respectively. (On Brick and Gravel, we attribute the high error to an overemphasis on high frequencies; see figure 7.) Even in these cases, the moment-matched initialization is no worse than mistuning ω_0 by 20%.

The baselines are all capable of achieving similar performance, but at ω_0 values that vary widely across images. For any given method and image, the loss landscape of ω_0 is nonconvex and contains local minima. The general trend is that SM20 performs best at the smallest ω_0, followed by CVP26 ($\sigma_a = 1$), followed by CVP26 ($\sigma_a = 0$). This ordering confirms an established finding (Combette et al., 2026) that SM20 is biased toward high frequencies and large gradients, followed by CVP26 ($\sigma_a = 1$) and CVP26 ($\sigma_a = 0$), in that order.

The situation differs when we fit in three-dimensional **CIELAB** color space, which is related to RGB space by a nonlinear transformation that approximates human perception. Perceived colors in CIELAB space are represented by $L^* \in [0, 100]$ and $a^*, b^* \in [-128, 127]$. This transformation preserves the spatial structure of the images but drastically alters the mean and (anisotropic) covariance of the color channels.

For all CIELAB images, our initialization achieves the best fit; the improvement is slight on Hubble and ranges from a factor of 3.0 to 5.2 on the other images. We attribute this to the fact that it correctly adapts to the distribution of the targets as well as to the typical resolution of spatial detail: the $\hat{\Omega}$ structure tensors of CIELAB images are very close to those of RGB images (Appendix E).

4.5 WIDTH SCALING USING μP: INITIALIZATION AND LEARNING RATES

In this section, we test how the optimal learning rate depends on width N. The universality theory of μP is based on three desiderata (Yang et al., 2022, Appendix J.2):

1. Every (pre)activation vector in a network should have $\Theta(1)$-sized coordinates.
2. Neural network output should be $O(1)$.
3. All parameters should be updated as much as possible (in terms of scaling in width) without leading to divergence.

Desideratum 1 is satisfied by the uniform-phase component of our initialization. Desideratum 2 is satisfied as $\Theta(1)$ by the moment-matching property of our initialization; this differs from μP's own last-layer prescription (Remark 1). Desideratum 3 concerns the learning rate, which is orthogonal to the focus of this paper; for the Adam optimizer, which we use, Yang et al. (2022) prescribe scaling the learning rate as $1/N$ for non-output layers. We test whether this prescription transfers the optimal learning rate across width for sinusoidal networks, and whether it interacts with our moment-matched initialization. (To our knowledge, no prior work has validated μP for either sinusoidal MLPs or the neural representation task.)

We repeat the experiment of section 4.1 for each of the four initializations at seven widths $N \in \{8, 16, 32, 64, 128, 256, 512\}$, using each initialization's best ω_0 from a width-64 sweep. At each width, we sweep the Adam learning rate over powers of two, $\eta \in \{2^{-18}, \ldots, 2^{2}\}$.

standard Plain Adam at the swept η with the initialization left unmodified.

μP (LR) The μP learning-rate scaling above; the initialization is unmodified.

μP (LR+LL) The same learning-rate scaling, with the last-layer weights also multiplied at initialization by $\sqrt{64/N}$.

Under standard scaling, the loss curve shifts left with width, and under both μP-informed scalings, the loss curve as a function of learning rate deepens with increasing width. These observations validate μP's predictions. But the dominant pattern is the presence of a phase transition immediately to the right of the optimal learning rate, where the loss dramatically increases to a null value (Figure 6). In both μP scalings, the phase transition and the optimal learning rate both shift right with width. While we have not observed optimal hyperparameter transfer, both μPs result in a loss that decreases with increasing width at every learning rate.

μP's last-layer rescaling in **LR+LL** does not appreciably differ from the **LR**-only scaling. This supports our theory that, as far as initialization is concerned, moment matching, which is architecture-agnostic, is as effective for learning-rate transfer as μP, which is architecture-aware.

5 CONCLUSION

Stable initialization of sinusoidal networks rests on two pillars: hidden layer stability and input-output layer tuning. Whereas the conventional theory for hidden layers requires subtle recursions and infinite-width analysis in order to control dependencies between neurons and weights, our method renders all preactivations independent and identically distributed, allowing network gradient moments to be obtained exactly by a short computation. We simplify the theory and give the strongest possible guarantee, one that holds for all widths and depths.

Whereas the conventional theory for input-output tuning relies on hand-tuning in the frequency domain, we take a physical-domain perspective on nondimensionalization: canceling the input and output scales using data-driven estimates of the neural field's covariance and structure tensor. The result is zero-shot initialization tuning on par with a grid search over prior hyperparameterizations.

Our work suggests that sine may be underappreciated as an activation function in machine learning, and that the uniform-phase property may prove fruitful in diverse applications beyond neural representation.

AI USE STATEMENT

(Required disclosure) **In this work, we used AI tools** to implement methods (replicate CVP26 and SM20, originally written in PyTorch, in JAX; implement our proposed initialization and evaluation harness). **We did not use AI tools** to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, or interpret results.

(Recommended disclosure) **Additionally, we used AI tools** to summarize or analyse existing literature, discover research topics or identify gaps, search for information, copyedit for grammaticality, and identify relevant literature.

We have reviewed all AI-assisted work. We carefully proofread AI-generated code for correctness to rule out bugs and reward-hacking tendencies. Furthermore, we have verified that our replications are within numerical tolerance of prior work. In the main body of the paper, our use of AI in writing assistance is strictly limited to technical copyediting; wording and stylistic choices are the product of human authorship. The problem description and reproducibility appendices are drafted by AI and checked for accuracy. We prepared the bibliography manually using a reference manager and have verified the relevance of each citation. **We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.**

ETHICS STATEMENT

We consider that this work does not raise questions respecting the Code of Ethics.

REPRODUCIBILITY STATEMENT

Appendix C specifies the data, preprocessing, estimators, initialization distributions, architectures, optimization, random seeds, metrics, and sweep grids used in every numerical experiment; the remaining appendices provide complete proofs and additional quantitative and visual results. The accompanying source archive includes the shared JAX/Equinox implementations in `src/architecture.py` and `src/moment_matching.py`; `src/cameraman-and-audio.ipynb` reproduces the image, narrow-network, and audio experiments; `src/omega-sweep.ipynb` and `src/mup-lr-sweep.ipynb` reproduce the two hyperparameter studies; `src/jacobian_svd_spectrum(CVP26comparison).ipynb` reproduces the spectral diagnostic; and `src/image_fitting-repro.ipynb` records the CVP26 image-fitting replication used to validate the shared implementation. The image data are obtained through `scikit-image`, and the Bach waveform is included at `src/assets/gt_bach.wav`. The two sweep notebooks are configured for accelerator-backed cloud execution and write to mounted `/mnt` directories; for local execution, the user must select an available JAX backend and change those output directories to writable local paths.

REFERENCES

Mohammed Alsakabi, Kejia Hu, John M. Dolan, and Ozan K. Tonguz. JA-SIREN: Deterministic Initialization for Sinusoidal Networks via Spectral Matching, June 2026. URL http://arxiv.org/abs/2606.06671. arXiv:2606.06671 [cs.CV] version: 1.

Filipe de Avila Belbute-Peres and J. Zico Kolter. Simple initialization and parametrization of sinusoidal networks via their kernel bandwidth. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=yVqC6gCNf4d.

Yizhak Ben-Shabat, Chamin Hewa Koneputugodage, and Stephen Gould. Digs: Divergence guided shape implicit neural representation for unoriented point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19323–19332, 2022.

Kyle R. Chickering. The Spectral Maximal Update Parameterization in Theory and Practice, 2025. URL https://kyrochi.github.io/blog/spectral-mup/index.html.

Andrea Combette, Nelly Pustelnik, and Antoine Venaille. A New Initialization to Control Gradients in Sinusoidal Neural Networks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=92d74WdgtG.

Nolan Dey, Bin Claire Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: CompleteP enables compute-efficient deep transformers, January 2026. URL http://arxiv.org/abs/2505.01618. arXiv:2505.01618 [cs.LG].

Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. COIN: COmpression with Implicit Neural representations, April 2021. URL http://arxiv.org/abs/2103.03123. arXiv:2103.03123 [eess.IV].

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Boris Hanin and David Rolnick. How to Start Training: The Effect of Initialization and Architecture, 2018. URL https://arxiv.org/abs/1803.01719. Version Number: 3.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, NY, 2009. ISBN 978-0-387-84857-0 978-0-387-84858-7. doi: 10.1007/978-0-387-84858-7. URL http://link.springer.com/10.1007/978-0-387-84858-7.

Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the Impact of the Activation function on Deep Neural Networks Training. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2672–2680. PMLR, May 2019. URL https://proceedings.mlr.press/v97/hayou19a.html.

Soufiane Hayou, Eugenio Clerico, Bobby He, George Deligiannidis, Arnaud Doucet, and Judith Rousseau. Stable ResNet. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pp. 1324–1332. PMLR, March 2021. URL https://proceedings.mlr.press/v130/hayou21a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, December 2015. doi: 10.1109/ICCV.2015.123. URL https://ieeexplore.ieee.org/document/7410480/.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep Learning Scaling is Predictable, Empirically, December 2017. URL http://arxiv.org/abs/1712.00409. arXiv:1712.00409 [cs.LG].

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. 2018. doi: 10.48550/ARXIV.1806.07572. URL https://arxiv.org/abs/1806.07572. Version Number: 4.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-Normalizing Neural Networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/5d44ee6f2c3f71b73125876103c8f6c4-Abstract.html.

Yann A. LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient BackProp. In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller (eds.), *Neural Networks: Tricks of the Trade*, volume 7700, pp. 9–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35288-1 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_3. URL http://link.springer.com/10.1007/978-3-642-35289-8_3. Series Title: Lecture Notes in Computer Science.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep Neural Networks as Gaussian Processes, 2017. URL https://arxiv.org/abs/1711.00165. Version Number: 3.

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, August 2020. URL http://arxiv.org/abs/2003.08934. arXiv:2003.08934 [cs.CV].

Dmytro Mishkin and Jiri Matas. All you need is a good init, February 2016. URL http://arxiv.org/abs/1511.06422. arXiv:1511.06422 [cs.LG].

Tiago Novello, Diana Aldana, Andre Araujo, and Luiz Velho. Tuning the Frequencies: Robust Training for Sinusoidal Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3071–3080, June 2025.

Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. Taming the waves: sine as activation function in deep neural networks, 2017. URL https://openreview.net/forum?id=Sks3zF9eg.

Jeffrey Pennington and Pratik Worah. The Spectrum of the Fisher Information Matrix of a Single-Hidden-Layer Neural Network. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/18bb68e2b38e4a8ce7cf4f6b2625768c-Abstract.html.

Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 30, 2017.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/148510031349642de5ca0c544f31b2ef-Paper.pdf.

Daniel A. Roberts, Sho Yaida, and Boris Hanin. *The Principles of Deep Learning Theory: An Effective Theory Approach to Understanding Neural Networks*. Cambridge University Press, 1 edition, May 2022. ISBN 978-1-009-02340-5 978-1-316-51933-2. doi: 10.1017/9781009023405. URL https://www.cambridge.org/core/product/identifier/9781009023405/type/book.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, February 2014. URL http://arxiv.org/abs/1312.6120. arXiv:1312.6120 [cs.NE].

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep Information Propagation, April 2017. URL http://arxiv.org/abs/1611.01232. arXiv:1611.01232 [stat.ML].

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/53c04118df112c13a8c34b38343b9c10-Abstract.html.

Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit Neural Representations for Image Compression. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, volume 13686, pp. 74–91. Springer Nature Switzerland, Cham, 2022. ISBN 978-3-031-19808-3 978-3-031-19809-0. doi: 10.1007/978-3-031-19809-0_5. URL https://link.springer.com/10.1007/978-3-031-19809-0_5. Series Title: Lecture Notes in Computer Science.

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, June 2020. URL http://arxiv.org/abs/2006.10739. arXiv:2006.10739 [cs.CV].

Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned Initializations for Optimizing Coordinate-Based Neural Representations, March 2021. URL http://arxiv.org/abs/2012.02189. arXiv:2012.02189 [cs.CV].

Pierre Wolinski and Julyan Arbel. Gaussian Pre-Activations in Neural Networks: Myth or Reality? 2022. doi: 10.48550/ARXIV.2205.12379. URL https://arxiv.org/abs/2205.12379. Version Number: 4.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S. Schoenholz, and Jeffrey Pennington. Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10,000-Layer Vanilla Convolutional Neural Networks, 2018. URL https://arxiv.org/abs/1806.05393. Version Number: 2.

Ge Yang and Samuel Schoenholz. Mean Field Residual Networks: On the Edge of Chaos. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/81c650caac28cdefce4de5ddc18befa0-Paper.pdf.

Greg Yang. Tensor Programs I: Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes, 2019. URL https://arxiv.org/abs/1910.12478. Version Number: 3.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer, March 2022. URL http://arxiv.org/abs/2203.03466. arXiv:2203.03466 [cs.LG].

Greg Yang, James B. Simon, and Jeremy Bernstein. A Spectral Condition for Feature Learning, May 2024. URL http://arxiv.org/abs/2310.17813. arXiv:2310.17813 [cs.LG].

Taesun Yeom, Sangyoon Lee, and Jaeho Lee. Fast training of sinusoidal neural fields via scaling initialization. In *International Conference on Learning Representations*, volume 2025, pp. 96229–96271, 2025.

Gizem Yuce, Guillermo Ortiz-Jimenez, Beril Besbinar, and Pascal Frossard. A Structured Dictionary Perspective on Implicit Neural Representations. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19206–19216, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-6946-3. doi: 10.1109/CVPR52688.2022.01863. URL https://ieeexplore.ieee.org/document/9878645/.

A Comparison to conventional stable initialization of hidden layers

We compare our hidden-layer initialization to that of Combette et al. (2026), which represents a state-of-the-art implementation of the conventional wisdom for stable initialization. Pursuant to the four-step organization described in the Introduction, there are two significant derivations.

A.1 Preactivation distribution

The first major effort is to approximate preactivations z_ℓ as asymptotically Normal with a tunable variance σ_a^2: $z_\ell \to \mathcal{N}(0, \sigma_a^2)$.

Prior work (Thm. 3.1, Combette et al. (2026)). *Consider the sinusoidal network defined above, where for some $c_w, c_b \in \mathbb{R}^+$, and for every layer $\ell \in \{2, \dots, L\}$, the weight matrix W_ℓ is initialized with entries sampled from $\mathcal{U}(-c_w/\sqrt{N},\, c_w/\sqrt{N})$, W_1 is sampled from $\mathcal{U}(-w_0/n_0,\, w_0/n_0)$, and the bias b_ℓ is initialized with entries sampled from $\mathcal{N}(0, c_b^2)$. Let $(z_\ell)_{\ell\in\{1,\dots,L\}}$ be the preactivation sequence $z_\ell = W_\ell x_{\ell-1} + b_\ell$ for an input $x \in \mathbb{R}^{n_0}$. Then, in the limits $N, L \to \infty$, the preactivation sequence $(z_\ell)_{\ell\in\mathbb{N}}$ converges in distribution to $\mathcal{N}(0, \sigma_a^2)$ with*

$$\sigma_a^2 = c_b^2 + \frac{c_w^2}{6} + \frac{1}{2}\mathcal{W}_0\left(-\frac{c_w^2}{3}e^{-\frac{c_w^2}{3}-2c_b^2}\right), \tag{13}$$

where $\mathcal{W}_0$ is the principal real branch of the Lambert W function. The sequence $\big(\mathrm{Var}(z_\ell)\big)_{\ell\in\mathbb{N}}$ converges to the fixed point σ_a^2, which is exponentially attractive for all $c_w \neq \sqrt{3}$. For $c_w = \sqrt{3}$, the convergence is of rate $\mathcal{O}(1/\ell)$.

The limit $N \to \infty$ is necessary in order to apply the Central Limit Theorem to establish that the preactivations converge weakly to an isotropic Normal distribution. (Because the sine activation and its derivatives are bounded and continuous, weak convergence is sufficient to reason about the network's moments.) The limit $L \to \infty$ is necessary in order to wash out the transient effect of the embedding layer on the preactivation distribution. An elegant nonlinear recursion is derived for each preactivation layer's moments in terms of those of the previous layer; this recursion has an attractive fixed point given by equation 13, reflecting a delicate balance between weights and biases.

All of this follows from the need to enforce a Normality Ansatz. While this Ansatz would make sense for nonperiodic activation functions, the parallel claim in our result is rendered nearly trivial by the fact that the sine function's domain can, without loss of generality, be wrapped onto a torus.

Our result. *Let b_ℓ be uniformly distributed on the N-torus. Let $z_\ell = W_\ell x_{\ell-1} + b_\ell$. Then $z_\ell \bmod 2\pi$ is uniformly distributed on the N-torus.*

Adding a uniform phase wipes out the distribution of $W_\ell x_{\ell-1}$. This eliminates the need for recursive analysis of the wrapped preactivation distribution because it is toroidally uniform by construction. There is also no need to take $L \to \infty$, because we have decoupled the layers and imposed an exactly uniform distribution on every layer's wrapped preactivation. Furthermore, there is no need to take $N \to \infty$, because we do not rely on the Central Limit Theorem. Our wrapped preactivation distribution is exact.

Finally, we observe that Combette et al. (2026, Thm. 3.1) is compatible with the key idea of our theory. Keeping c_w finite, take the limit $c_b^2 \to \infty$. Using $\mathcal{W}_0(0) = 0$, equation 13 reduces in this limit to

$$\sigma_a^2 = c_b^2 = \infty. \tag{14}$$

When a Normal distribution with infinite variance is wrapped onto the torus, the result is a uniform distribution, recovering our finding that the preactivation distribution can be decoupled from the weight scale c_w.

A.2 Activation gradient distribution

Using the preactivation distribution, one can now estimate the quadratic mean of the activation function's slope over the preactivation ensemble through the more general identity

$$\mathrm{Var}\left[\frac{\partial}{\partial z_\ell}\sin(z_\ell)\right] = \mathbb{E}\left[\cos^2(z_\ell)\right] = \frac{1 + e^{-2\sigma_a^2}}{2} \quad \text{elementwise.} \tag{15}$$

This principle may be used to analyze the layer Jacobian entrywise:

Prior work (Thm. 3.2, Combette et al. (2026)). *Let $\partial x_\ell/\partial x_{\ell-1}$ denote the Jacobian of the ℓ-th layer, so that*

$$\frac{\partial x_\ell}{\partial x_{\ell-1}} = \mathrm{diag}(\cos(z_\ell))\, W_\ell. \tag{16}$$

Under the same assumptions as above, and maintaining the limit of large N, each entry of this Jacobian has zero mean and variance $\widetilde{\sigma}_\ell^2$, such that the sequence $(N\widetilde{\sigma}_\ell^2)_{\ell\in\mathbb{N}}$ converges to

$$\lim_{\ell, N\to\infty}(N\widetilde{\sigma}_\ell^2) = \sigma_g = \frac{c_w^2}{6}\big(1 + e^{-2\sigma_a^2}\big). \tag{17}$$

A parallel result can also be stated for our initialization:

Our result. *Let the wrapped preactivation $z_\ell \bmod 2\pi$ be uniformly distributed on the N-torus. Assume that each entry of W_ℓ has zero mean and variance $c_w^2/(3N)$. Then each entry of the Jacobian*

$$\frac{\partial x_\ell}{\partial x_{\ell-1}} = \mathrm{diag}(\cos(z_\ell))W_\ell \tag{18}$$

has zero mean and variance $c_w^2/(6N)$.

Note again the lack of asymptotics. Activation-slope vectors such as $\cos(z_\ell)$ are i.i.d.

A.3 What gradient criterion?

Finally, we turn to the choice of criterion to stabilize (Step 1 of the Introduction). Whereas Combette et al. (2026) stabilize the *entries* of the Jacobian, we stabilize its gramian matrix. Translated into our convention, their result (Appendix A.4) may be summarized as follows.

Prior work (Appendix A.4, Combette et al. (2026))**.** *Let f be a sinusoidal network initialized so that each entry of the layerwise Jacobian $\partial x_\ell/\partial x_{\ell-1}$ has variance σ_g^2/N. Let Ψ be a scalar loss function applied to the network output. Then the parameter-wise and input-wise gradient variances scale as*

$$\mathrm{Var}\left[\nabla_{W_{\ell,i,j}}\Psi(f(x))\right] \sim \frac{(N\sigma_g^2)^{L-\ell-1}}{N}, \qquad \ell > 1, \tag{19}$$

$$\mathrm{Var}\left[\nabla_{x_i}\Psi(f(x))\right] \sim (\sigma_g^2)^{L-2}. \tag{20}$$

In particular, gradients vanish or explode exponentially with depth unless $N\sigma_g^2 \approx 1$.

This scaling analysis is a plausible—if not fully rigorous (it applies an unjustified independence heuristic to random matrix-vector products)—corollary of the elementwise Jacobian result of Combette et al. (2026, Thm. 3.2). However, it relies on the double limit $\ell, N \to \infty$ and hides an unknown constant.

Our initialization removes the asymptotic limits from the Jacobian second-moment calculation. Translating that calculation into a loss-gradient prediction requires an additional assumption on the loss cotangent.

Our result. *Let f be a sinusoidal network with the uniform-phase initialization, let $2 \le \ell \le L$, and let $g = \nabla_{x_L}\Psi(f(x)) \in \mathbb{R}^N$ denote the loss cotangent at the output of the hidden map. Then the parameter and input gradients decompose as*

$$\nabla_{W_\ell}\Psi(f(x)) = \mathrm{Cos}(z_\ell)\left(\frac{\partial x_L}{\partial x_\ell}\right)^\intercal g x_{\ell-1}^\intercal \quad \textit{and} \tag{21}$$

$$\nabla_{x_0}\Psi(f(x)) = \left(\frac{\partial x_L}{\partial x_0}\right)^\intercal g. \tag{22}$$

Using an independence heuristic,

$$\mathbb{E}\left(\nabla_{W_\ell}\Psi(f(x))\right)^\intercal\left(\nabla_{W_\ell}\Psi(f(x))\right) \approx \frac{1}{4}\|g\|^2 I \quad \textit{and} \tag{23}$$

$$\mathbb{E}\left(\nabla_{x_0}\Psi(f(x))\right)^\intercal\left(\nabla_{x_0}\Psi(f(x))\right) \approx \|g\|^2. \tag{24}$$

Proof of equation 23. Most of the independence we need is rigorous (Lemma 1). The remaining approximation treats g as approximately independent of the hidden-layer initialization (Chickering, 2025; Yang et al., 2024). As in Combette et al. (2026), this is an early-training heuristic: for a general

loss, the cotangent depends on the initialized network.

$$\begin{aligned}\mathbb{E}\left(\nabla_{W_\ell}\Psi(f(x))\right)^\intercal\left(\nabla_{W_\ell}\Psi(f(x))\right) &= \mathbb{E}\, x_{\ell-1}g^\intercal\left(\frac{\partial x_L}{\partial x_\ell}\right)\mathrm{Cos}(z_\ell)\,\mathrm{Cos}(z_\ell)\left(\frac{\partial x_L}{\partial x_\ell}\right)^\intercal g x_{\ell-1}^\intercal \\ &\approx \frac{1}{2}\left\|g\right\|^2\mathbb{E}\, x_{\ell-1}x_{\ell-1}^\intercal \\ &\approx \frac{1}{2}\left\|g\right\|^2\mathbb{E}\sin(z_{\ell-1})\sin(z_{\ell-1})^\intercal \\ &\approx \frac{1}{4}\left\|g\right\|^2 I\end{aligned}$$

□

A.4 Spectral diagnostics

We complement the theory of Section 2 with mechanistic diagnostics that illuminate how the uniform-phase initialization differs from its predecessors: the singular value spectrum of the end-to-end Jacobian and the visual appearance of random initialization fields. Although Theorem 1 and CVP26's asymptotic analysis make different predictions about the distribution of singular values, neither fully determines the spectral shape. Therefore we inspect it empirically.

We extend Combette et al. (2026, Figure 8), which plots the singular values of $J = \dfrac{\partial f}{\partial x}$ from greatest to least, by adding the uniform-phase initialization. To ensure a fair comparison, the input and output layers of our moment-matched uniform-phase initialization are calibrated to match the output statistics of a CVP26 ($\sigma_a = 0$, $L = 4$) network, so that the only variable is the hidden-layer initialization.

The uniform-phase initialization exhibits spectral broadening with depth: the largest singular values grow ("rich get richer") while the smallest decay toward zero ("poor get poorer"). This is a way of satisfying Theorem 1, which fixes the *average* squared singular value at unity while leaving the distribution's shape unconstrained. If some singular values grow, others must shrink to balance the mean.

This contrasts sharply with CVP26's $\sigma_a = 0$ initialization, which instead preserves the *largest* singular value (the operator norm) across depth. There, the "poor get poorer" phenomenon occurs for a different reason: the operator norm is held fixed while the average squared singular value decays, so the network's mean amplification shrinks even as its peak amplification holds steady. The two methods thus embody different design goals: CVP26 σ_a=0 stabilizes the worst case, while the uniform-phase initialization stabilizes the average case.

The mechanism is also very different. CVP26 concentrates preactivations on the steep portion of the sine, amplifying slopes uniformly, whereas the uniform-phase initialization averages over the full sine (including its stationary and downward-sloping regions). The effect is most visible in comparison to SM20, which uses the same weight distribution as the uniform-phase initialization but smaller biases, and consequently suffers gradient blowup.

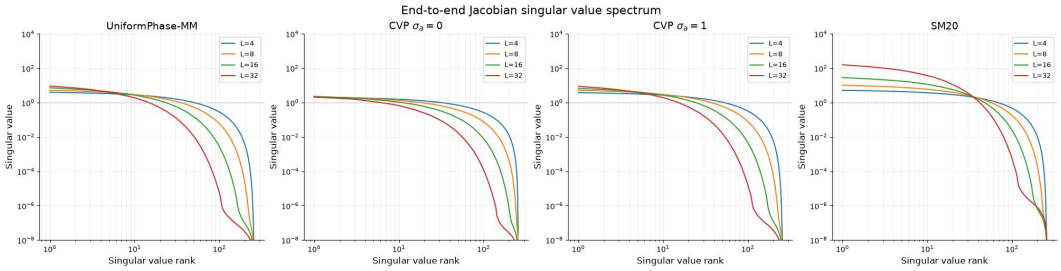

Figure 3: Full singular value spectrum of the end-to-end Jacobian $\mathbf{J} = \partial \mathbf{h}_L / \partial \mathbf{h}_1$ as a function of depth. Four initialization schemes are compared: uniform-phase MM (moment-matched to CVP26 σ_a=0, L=4), CVP26 σ_a=0, CVP26 σ_a=1, and the original SM20 initialization. Each spectrum is averaged over five independently initialized networks, with 10 sample points on $[-\pi, \pi]$.

B SUPPLEMENT TO §4.5

Remark 1 (μP last-layer scaling)**.** *Whereas moment matching scales the last-layer weights by $N^{-1/2}$ (§4.5), μP insists on scaling them by N^{-1}, which causes the network output to decay as $N^{-1/2}$. See Yang et al. (2022, Appendix J.2.1) for the derivation, and Appendix D.2, ibid., for a justification of output weight collapse. We note in passing that last-layer weights are intuitively easier to learn because training the last layer while holding other parameters fixed is a convex problem; nevertheless, in empirical training, the representation learning task can be sensitive to output scaling.*

C TASKS AND EXPERIMENTAL METHODS

Common protocol and metrics. The fitting experiments represent a sampled field with a coordinate MLP and minimize mean squared error (MSE) over all samples and output dimensions. Every optimizer step is full-batch. Model parameters and training arrays use 32-bit floating-point values, while moment estimates are accumulated in 64-bit NumPy and then converted to model precision. Adam is used without weight decay, gradient clipping, or a learning-rate schedule. For image and audio fitting we report

$$\mathrm{SNR} = 10 \log_{10} \frac{\mathrm{Var}(y)}{\mathrm{MSE}(\hat{y}, y)}, \tag{25}$$

where the variance, like the MSE, is computed over the complete evaluation array. The high-resolution image grid and the full audio signal are evaluation sets rather than independently sampled observations. We present the image artifacts with matplotlib-default lossy compression in order to reduce the size of this file, and certify that it does not materially affect their interpretation.

Initialization implementations. For uniform-phase MM, let a regularly sampled training field contain m values $y_i \in \mathbb{R}^{N_{\mathrm{out}}}$ and let J_i denote its spatial Jacobian. We use the sample mean and covariance and form

$$\widehat{\Sigma}_\epsilon = \widehat{\Sigma} + \epsilon I, \qquad \widehat{\Omega} = \frac{1}{m} \sum_{i=1}^{m} J_i^\intercal \widehat{\Sigma}_\epsilon^{-1} J_i + \epsilon I, \qquad \epsilon = 10^{-6}. \tag{26}$$

Each J_i is estimated channel by channel with `scipy.ndimage.sobel`; the result is divided by the coordinate spacing and by $2 \cdot 4^{d-1}$, which accounts for the centered difference and smoothing in the other $d-1$ spatial directions. The input weights have independent rows distributed as $\mathcal{N}(0, \widehat{\Omega}/N_{\mathrm{out}})$, the input phase and every hidden bias are independent $\mathcal{U}(-\pi, \pi)$ variables, and hidden weights are independent $\mathcal{N}(0, 2/N)$ variables. The output bias is $\widehat{\mu}$ and the columns of the output matrix are independent $\mathcal{N}(0, 2\widehat{\Sigma}_\epsilon/N)$ variables.

The SM20 and CVP26 implementations follow the corresponding PyTorch releases (Sitzmann et al., 2020; Combette et al., 2026) but are written in JAX/Equinox. Their first-layer weights are sampled uniformly on $[-1/N_{\mathrm{in}}, 1/N_{\mathrm{in}}]$ and then multiplied by the reported first-layer ω_0 inside the sine. For

subsequent sine layers, $W_{ij} \sim \mathcal{U}[-c/(\omega_h\sqrt{N}), c/(\omega_h\sqrt{N})]$ and the activation is $\sin(\omega_h(Wx+b))$. SM20 uses $c = \sqrt{6}$ and PyTorch-style uniform biases; CVP26 $\sigma_a = 0$ uses $c = \sqrt{3}$ and zero hidden biases; and CVP26 $\sigma_a = 1$ uses $c = \sqrt{6/(1+e^{-2})}$ and centered Normal hidden biases of variance $c^2e^{-2}/3$ when $\omega_h = 1$. The final linear map has no bias and has weights uniform on $[-\sqrt{3/N}/\omega_h, \sqrt{3/N}/\omega_h]$. The Tanh (PE-Xavier) baseline encodes x as $[\sin(2\pi Bx), \cos(2\pi Bx)]$, using $N/2$ rows of B sampled independently from $\mathcal{N}(0, \sigma^2 I)$, and applies a Xavier-initialized Tanh MLP. The ReLU baseline uses Kaiming-uniform weights, while the SiLU and GeLU baselines use Xavier-uniform weights; these MLPs use zero biases. All baselines match the compared SIRENs in width and depth.

Cameraman image fitting and narrow networks. We take `skimage.data.camera`, downsample it from its native 512×512 resolution to 128×128 with anti-aliasing, and associate pixels with an endpoint-inclusive Cartesian grid on $[-1, 1]^2$. Training uses all 16,384 low-resolution coordinate–value pairs; evaluation uses all 262,144 native-resolution pairs. Pixel intensities are represented in $[0, 1]$. The standard experiment uses the architecture and hyperparameters of Combette et al. (2026)'s supplementary material: $L = 11$, $N = 256$, 3000 Adam steps, and learning rate 10^{-4}; the narrow experiment changes only the width to $N = 16$. The first-layer frequency is $64\pi/\sqrt{6}$ for SM20 and CVP26, and hidden frequencies are 1. The positional-encoding scale is $\sigma = 32$.

Audio fitting. The bundled `gt_bach.wav` file is loaded at its native 44.1 kHz sampling rate, converted to mono, and normalized by its maximum absolute amplitude. It contains 308,207 samples (approximately 7 s), which are placed on an endpoint-inclusive grid on $[-100, 100]$. The full signal is used both to optimize and to compute MSE and SNR; there is no held-out or subsampled split. All models use $L = 5$, $N = 256$, 5000 full-batch Adam steps, and learning rate 5×10^{-5}. Both baseline SIRENs use first-layer $\omega_0 = 30$; CVP26 uses hidden $\omega_h = 1$, whereas the SM20 replication uses $\omega_h = 30$ as in its reference implementation. The Tanh positional-encoding scale is $\sigma = 10$.

Input-frequency sweep. For each image and for each of SM20, CVP26 $\sigma_a = 0$, and CVP26 $\sigma_a = 1$, we evaluate

$$\omega_0 = \frac{64\pi}{\sqrt{6}}\, 10^q, \qquad q \in \{-2, -1.8, \ldots, 1.8, 2\}. \tag{27}$$

A baseline uses the same random weights at all 21 frequencies, so frequency is the only within-method variable; uniform-phase MM is initialized and trained once per image. All runs use $L = 11$, $N = 256$, and 3000 full-batch Adam steps at learning rate 10^{-4}, with training and evaluation grids whose largest dimensions are 128 and 512, respectively, and whose aspect ratio is preserved. The grayscale tasks use `brick`, `camera`, `gravel`, and `clock`; the color tasks use `hubble_deep_field`, `skin`, `astronaut`, and `coffee` from `scikit-image`. RGB values are scaled to $[0, 1]$. For the CIELAB variant, images are resized in RGB and then transformed with `skimage.color.rgb2lab`; training and MSE evaluation take place in CIELAB, and conversion back to RGB is used only for display.

Width and learning-rate sweep. The μP experiment uses the grayscale Cameraman training field and reports no test loss. First, at base width $N_0 = 64$, each prior SIREN initialization is trained at the 21 frequencies above using a learning rate of 10^{-4}; its frequency is selected by the mean of the 3000 pre-update training losses. Uniform-phase MM requires no such selection. Second, for $N \in \{8, 16, 32, 64, 128, 256, 512\}$, each initialization is trained at each $\eta \in \{2^{-18}, 2^{-17}, \ldots, 2^2\}$ for 3000 steps. The plotted criterion is again the time average of the per-step pre-update training MSE, rather than final MSE. All learning rates for a given initialization and width begin from identical parameters. In this experiment Adam uses $\epsilon = 0$ so that its additive numerical constant does not introduce a width-dependent scale. Under **μP (LR)**, hidden matrices and the output projection use an effective learning rate of $\eta N_0/N$, while input projections and all biases use η. Under **μP (LR+LL)**, the same optimizer scaling is used and the initialized output weights are additionally multiplied by $\sqrt{N_0/N}$. The **standard** condition uses η for every parameter and leaves initialization unchanged.

Jacobian-spectrum diagnostic. Following the protocol of Combette et al. (2026, Appendix B.1), we study the hidden-to-hidden Jacobian $J = \partial h_L/\partial h_1$ for width $N = 256$ and depths $L \in \{4, 8, 16, 32\}$. The uniform-phase input and output moments are estimated from evaluations of

a CVP26 $\sigma_a = 0$, $L = 4$ network with first-layer $\omega_0 = 30$ and hidden $\omega_h = 1$ at 1000 evenly spaced points on $[-\pi, \pi]$. For each initialization and depth, we compute sorted singular values at 10 uniformly sampled input points for each of five independently initialized networks and average each ordered singular value over the resulting 50 spectra.

D ADDITIONAL EXPERIMENTAL RESULTS

Table 1: Quantitative results for the Cameraman image-fitting experiment (§4.1). Train MSE is computed on the 128×128 grid; eval MSE and SNR are computed on the 512×512 grid. Best values are in bold.

Network	Train MSE	Eval MSE	SNR (dB)
SIREN (uniform-phase MM)	**0.000032**	**0.002706**	**14.89**
SIREN (CVP26 $\sigma_a = 0$)	0.000127	0.002967	14.49
SIREN (SM20)	0.000000	0.017428	6.80
ReLU (Kaiming)	0.001768	0.005201	12.05
SiLU (Xavier)	0.007591	0.011650	8.55
GeLU (Xavier)	0.010906	0.015013	7.45
Tanh (PE-Xavier)	0.000274	0.124271	−1.73

Table 2: Quantitative results for the narrow ($N = 16$) Cameraman experiment (§4.1).

Network	Train MSE	Eval MSE	SNR (dB)
SIREN (uniform-phase MM)	**0.004117**	**0.007824**	**10.28**
ReLU (Kaiming)	0.006713	0.010689	8.92
GeLU (Xavier)	0.017129	0.021263	5.94
SiLU (Xavier)	0.018928	0.023064	5.58
SIREN (CVP26 $\sigma_a = 0$)	0.033747	0.037765	3.44
SIREN (SM20)	0.062279	0.068359	0.86
Tanh (PE-Xavier)	0.073832	0.088348	−0.25

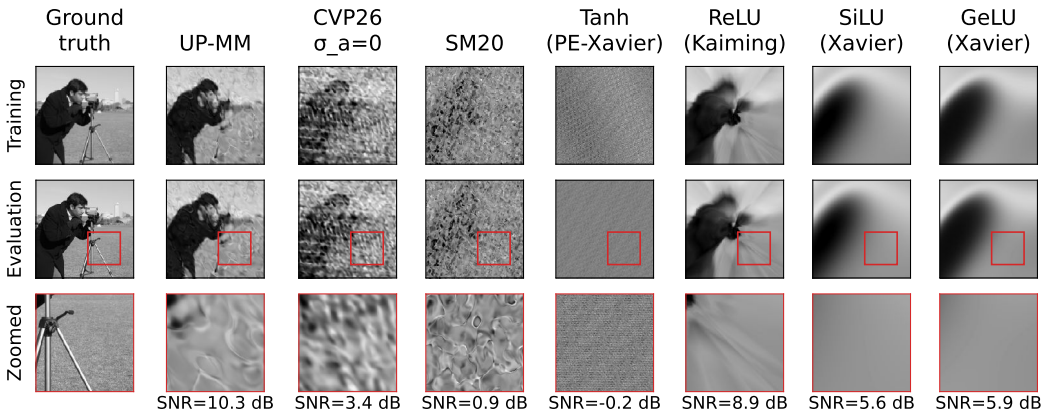

Figure 4: Image fitting with narrow networks ($N = 16$). Layout as in figure 1.

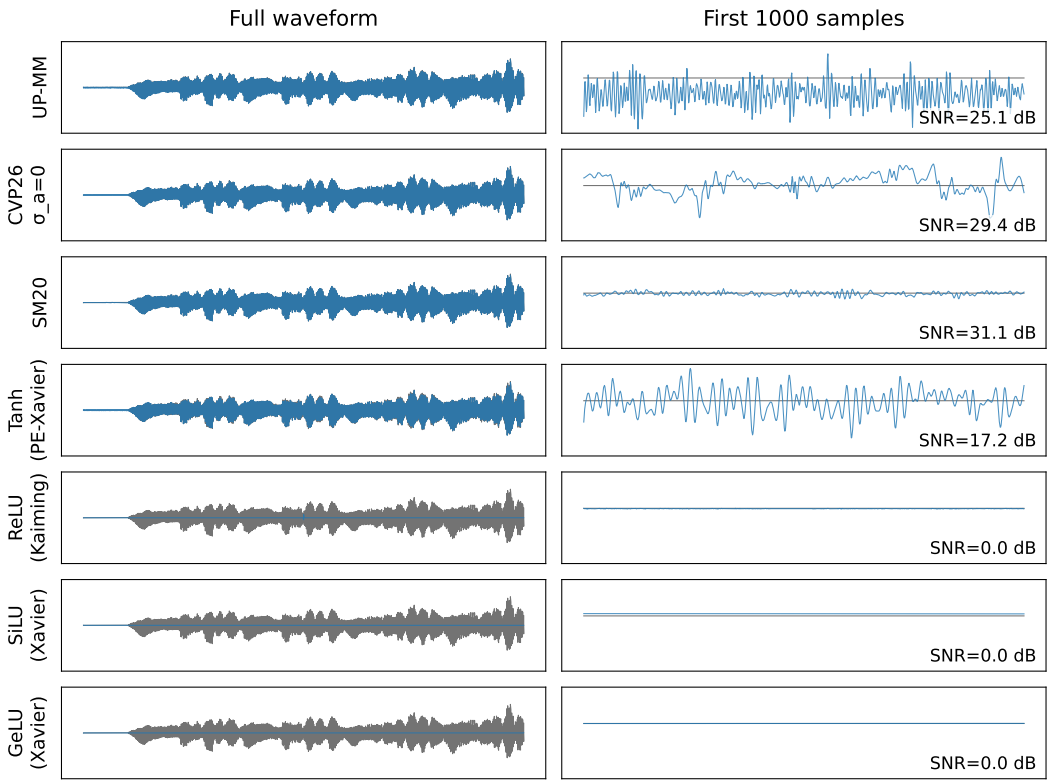

Figure 5: Audio fitting on the Bach recording. One row per initialization, with the ground truth in gray and the prediction in blue. Left: the full waveform, drawn as a per-pixel minimum/maximum envelope of all 308,207 samples. Right: the first 1000 samples, on its own vertical scale because the recording opens in near-silence. The evaluation SNR is given in each right-hand panel.

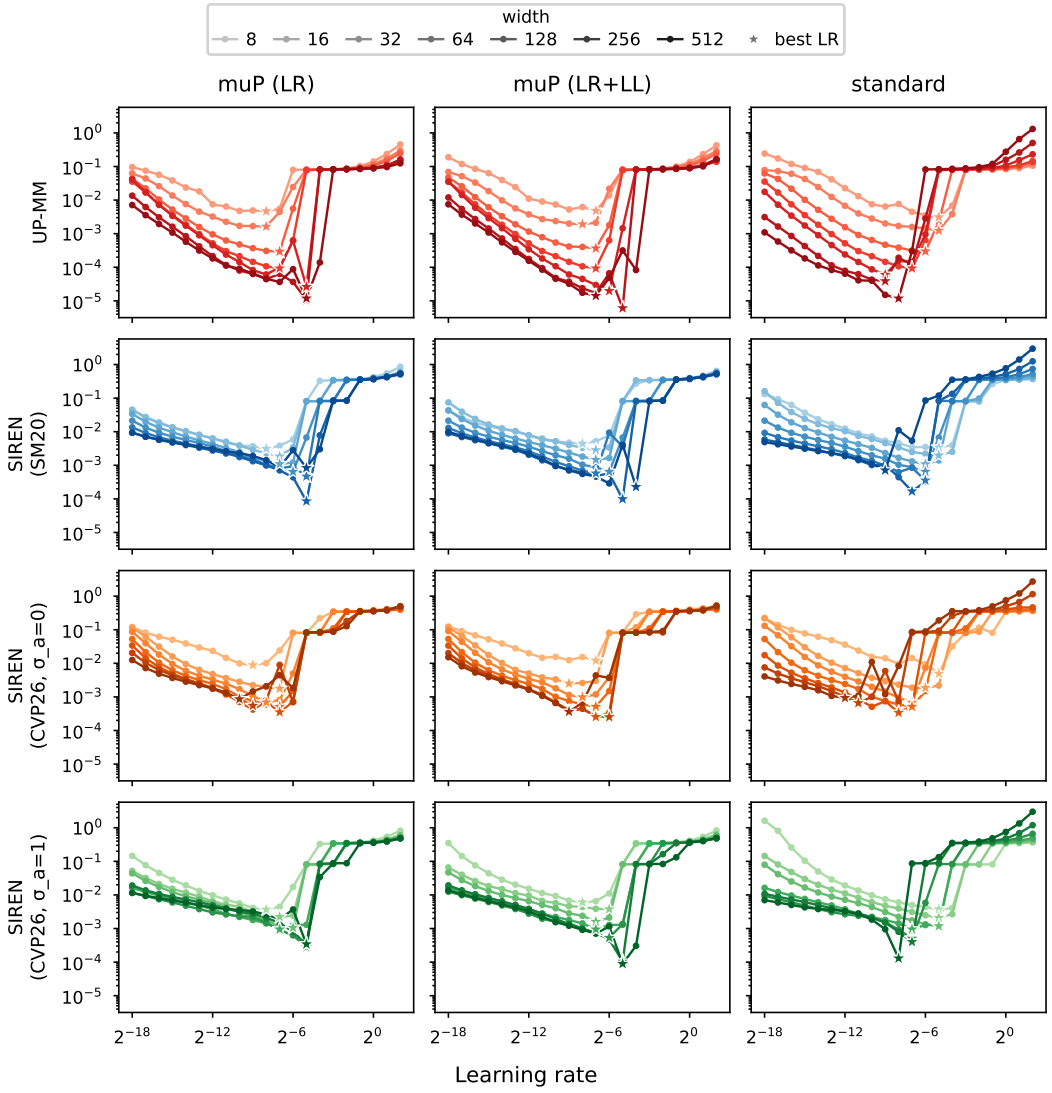

Figure 6: μP learning-rate transfer, grayscale Cameraman, geometrically time-averaged train MSE (§4.5). Rows: initialization (uniform-phase MM and the three SIREN baselines at their best ω_0); columns: μP LR, μP LR+LL, and standard scaling. One curve per width, shaded from light ($N = 8$) to dark ($N = 512$), with a star at each width's best learning rate.

E MOMENT-MATCHED PARAMETERS AND BEST-ω_0 TEST ERRORS

Grayscale The estimated moment-matched parameters are:

$$\hat{\mu} = 0.437 \qquad \hat{\Sigma} = 0.00672 \qquad \hat{\Omega} = \begin{pmatrix} 561 & -4.2 \\ -4.2 & 1870 \end{pmatrix} \quad \text{(Brick)}$$

$$\hat{\mu} = 0.506 \qquad \hat{\Sigma} = 0.0793 \qquad \hat{\Omega} = \begin{pmatrix} 87.0 & 10.9 \\ 10.9 & 115 \end{pmatrix} \quad \text{(Camera)}$$

$$\hat{\mu} = 0.496 \qquad \hat{\Sigma} = 0.0125 \qquad \hat{\Omega} = \begin{pmatrix} 1204 & -48.6 \\ -48.6 & 1120 \end{pmatrix} \quad \text{(Gravel)}$$

$$\hat{\mu} = 0.574 \qquad \hat{\Sigma} = 0.00664 \qquad \hat{\Omega} = \begin{pmatrix} 56.5 & 0.53 \\ 0.53 & 30.7 \end{pmatrix} \quad \text{(Clock)}$$

Table 3: Grayscale sweep: best test MSE per method. The optimal ω_0 from the grid search is shown in parentheses. Bold entries are the best per row. Visual comparisons: figure 7.

Image	uniform-phase MM	SM20	CVP26 σ_a=0	CVP26 σ_a=1
Brick	0.002852	**0.002317** (ω=20.6)	0.002408 (ω=82.1)	0.002381 (ω=32.7)
Camera	**0.002695**	0.002763 (ω=20.6)	0.002912 (ω=130)	0.002855 (ω=51.8)
Gravel	0.009063	0.007982 (ω=32.7)	0.007994 (ω=130)	**0.007937** (ω=51.8)
Clock	**0.000036**	0.000055 (ω=20.6)	0.000050 (ω=82.1)	0.000042 (ω=20.6)

Color (RGB) The estimated moment-matched parameters are:

$$\hat{\mu} = \begin{pmatrix} 0.073 \\ 0.078 \\ 0.075 \end{pmatrix} \quad \hat{\Sigma} = \begin{pmatrix} 0.0068 & 0.0054 & 0.0056 \\ 0.0054 & 0.0047 & 0.0050 \\ 0.0056 & 0.0050 & 0.0054 \end{pmatrix} \quad \hat{\Omega} = \begin{pmatrix} 2323 & -14.8 \\ -14.8 & 3010 \end{pmatrix} \quad \text{(Hubble)}$$

$$\hat{\mu} = \begin{pmatrix} 0.809 \\ 0.660 \\ 0.743 \end{pmatrix} \quad \hat{\Sigma} = \begin{pmatrix} 0.0031 & 0.0052 & 0.0030 \\ 0.0052 & 0.0165 & 0.0095 \\ 0.0030 & 0.0095 & 0.0059 \end{pmatrix} \quad \hat{\Omega} = \begin{pmatrix} 699 & 23.9 \\ 23.9 & 888 \end{pmatrix} \quad \text{(Skin)}$$

$$\hat{\mu} = \begin{pmatrix} 0.555 \\ 0.415 \\ 0.378 \end{pmatrix} \quad \hat{\Sigma} = \begin{pmatrix} 0.097 & 0.075 & 0.066 \\ 0.075 & 0.084 & 0.082 \\ 0.066 & 0.082 & 0.086 \end{pmatrix} \quad \hat{\Omega} = \begin{pmatrix} 551 & -56.1 \\ -56.1 & 804 \end{pmatrix} \quad \text{(Astronaut)}$$

$$\hat{\mu} = \begin{pmatrix} 0.622 \\ 0.336 \\ 0.202 \end{pmatrix} \quad \hat{\Sigma} = \begin{pmatrix} 0.057 & 0.046 & 0.032 \\ 0.046 & 0.052 & 0.042 \\ 0.032 & 0.042 & 0.038 \end{pmatrix} \quad \hat{\Omega} = \begin{pmatrix} 400 & 10.1 \\ 10.1 & 630 \end{pmatrix} \quad \text{(Coffee)}$$

Table 4: Color sweep: best test MSE per method. Conventions as in Table 3. Visual comparisons: figure 8.

Image	uniform-phase MM	SM20	CVP26 σ_a=0	CVP26 σ_a=1
Hubble	0.003773	0.003758 (ω=32.7)	**0.003715** (ω=130)	0.003808 (ω=51.8)
Skin	**0.005784**	0.006044 (ω=20.6)	0.006083 (ω=130)	0.005870 (ω=51.8)
Astronaut	**0.004027**	0.004076 (ω=32.7)	0.004276 (ω=130)	0.004191 (ω=51.8)
Coffee	**0.002728**	0.002785 (ω=20.6)	0.002847 (ω=82.1)	0.002874 (ω=51.8)

Color (CIELAB) The estimated moment-matched parameters in CIELAB space are:

$$\hat{\mu} = \begin{pmatrix} 6.47 \\ -0.35 \\ 0.16 \end{pmatrix} \quad \hat{\Sigma} = \begin{pmatrix} 64.4 & 12.3 & -0.21 \\ 12.3 & 7.55 & 2.56 \\ -0.21 & 2.56 & 5.50 \end{pmatrix} \quad \hat{\Omega} = \begin{pmatrix} 2328 & -9.7 \\ -9.7 & 3023 \end{pmatrix} \quad \text{(Hubble)}$$

$$\hat{\mu} = \begin{pmatrix} 72.99 \\ 17.52 \\ -5.39 \end{pmatrix} \quad \hat{\Sigma} = \begin{pmatrix} 90.97 & -99.67 & 36.77 \\ -99.67 & 130.15 & -39.38 \\ 36.77 & -39.38 & 24.02 \end{pmatrix} \quad \hat{\Omega} = \begin{pmatrix} 728 & 27.0 \\ 27.0 & 936 \end{pmatrix} \quad \text{(Skin)}$$

$$\hat{\mu} = \begin{pmatrix} 47.73 \\ 13.55 \\ 11.95 \end{pmatrix} \quad \hat{\Sigma} = \begin{pmatrix} 843 & -0.11 & 53.5 \\ -0.11 & 302 & 235 \\ 53.5 & 235 & 317 \end{pmatrix} \quad \hat{\Omega} = \begin{pmatrix} 571 & -54.0 \\ -54.0 & 822 \end{pmatrix} \quad \text{(Astronaut)}$$

$$\hat{\mu} = \begin{pmatrix} 44.39 \\ 26.60 \\ 32.84 \end{pmatrix} \quad \hat{\Sigma} = \begin{pmatrix} 495 & -36.95 & 126.4 \\ -36.95 & 189 & 145.3 \\ 126.4 & 145.3 & 203.3 \end{pmatrix} \quad \hat{\Omega} = \begin{pmatrix} 418 & 8.58 \\ 8.58 & 646 \end{pmatrix} \quad \text{(Coffee)}$$

Compared with the RGB parameters above, the CIELAB output covariance $\hat{\Sigma}$ has much larger entries (reflecting the wider dynamic range of CIELAB coordinates) and mixed-sign off-diagonal entries (reflecting the decorrelating effect of the opponency axes). The structure tensor $\hat{\Omega}$ is nearly unchanged, confirming that the spatial-frequency content is preserved under the color-space transformation.

Table 5: CIELAB sweep: best test MSE per method. MSE is computed in CIELAB coordinates. Conventions as in Table 3. Visual comparisons: figure 9.

Image	uniform-phase MM	SM20	CVP26 σ_a=0	CVP26 σ_a=1
Hubble	**18.148**	18.505 (ω=13.0)	18.675 (ω=82.1)	18.692 (ω=51.8)
Skin	**45.172**	134.00 (ω=13.0)	145.37 (ω=8208)	141.75 (ω=8.21)
Astronaut	**20.011**	83.61 (ω=32.7)	72.30 (ω=82.1)	84.91 (ω=20.6)
Coffee	**15.658**	81.19 (ω=20.6)	82.15 (ω=32.7)	88.28 (ω=32.7)

F BEST-ω_0 VISUAL COMPARISONS

Each figure in this section covers one experiment: rows are the four images, and columns are the ground truth, the uniform-phase MM fit, and the best-tuned (by ω_0) fits of SM20, CVP26 σ_a=0, and CVP26 σ_a=1 from the sweep of Appendix E. Each panel is annotated with its test MSE and, for the baselines, the ω_0 that achieved it.

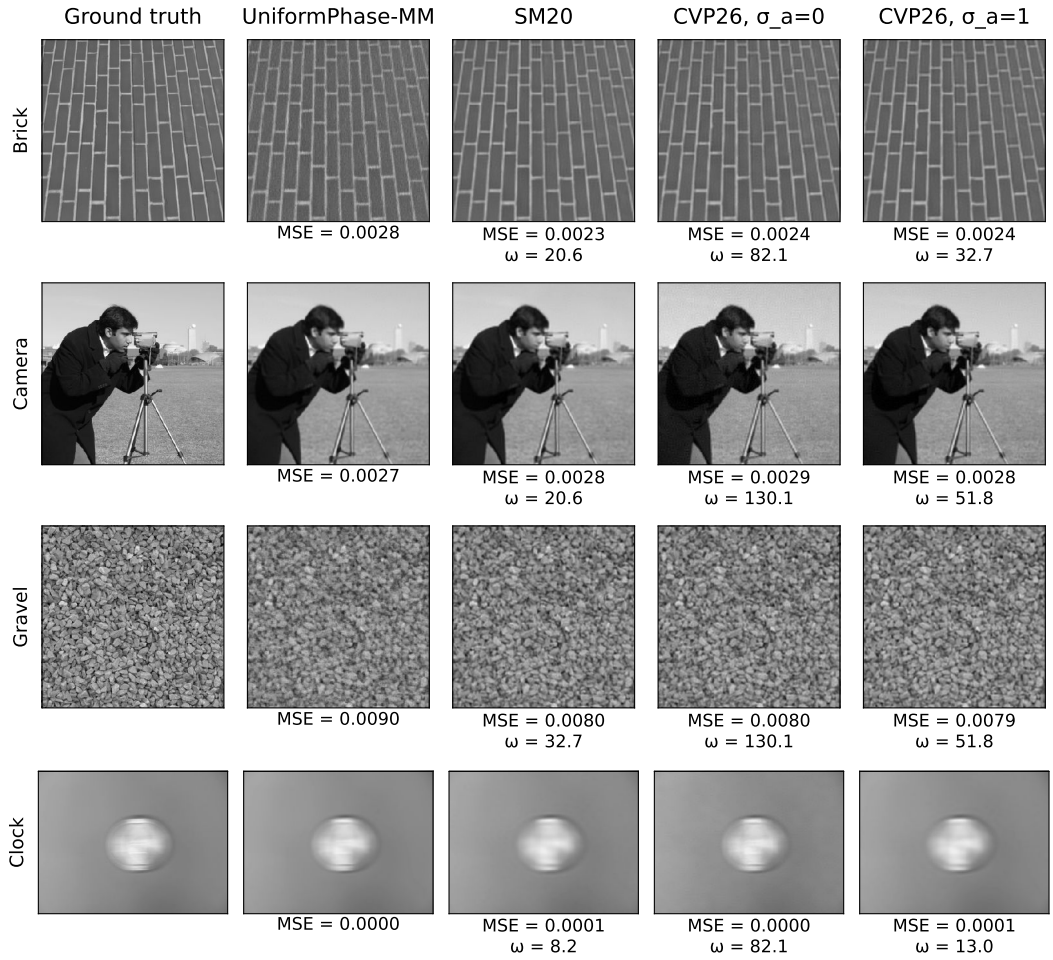

Figure 7: Grayscale best-ω_0 visual comparisons: Brick, Camera, Gravel, and Clock.

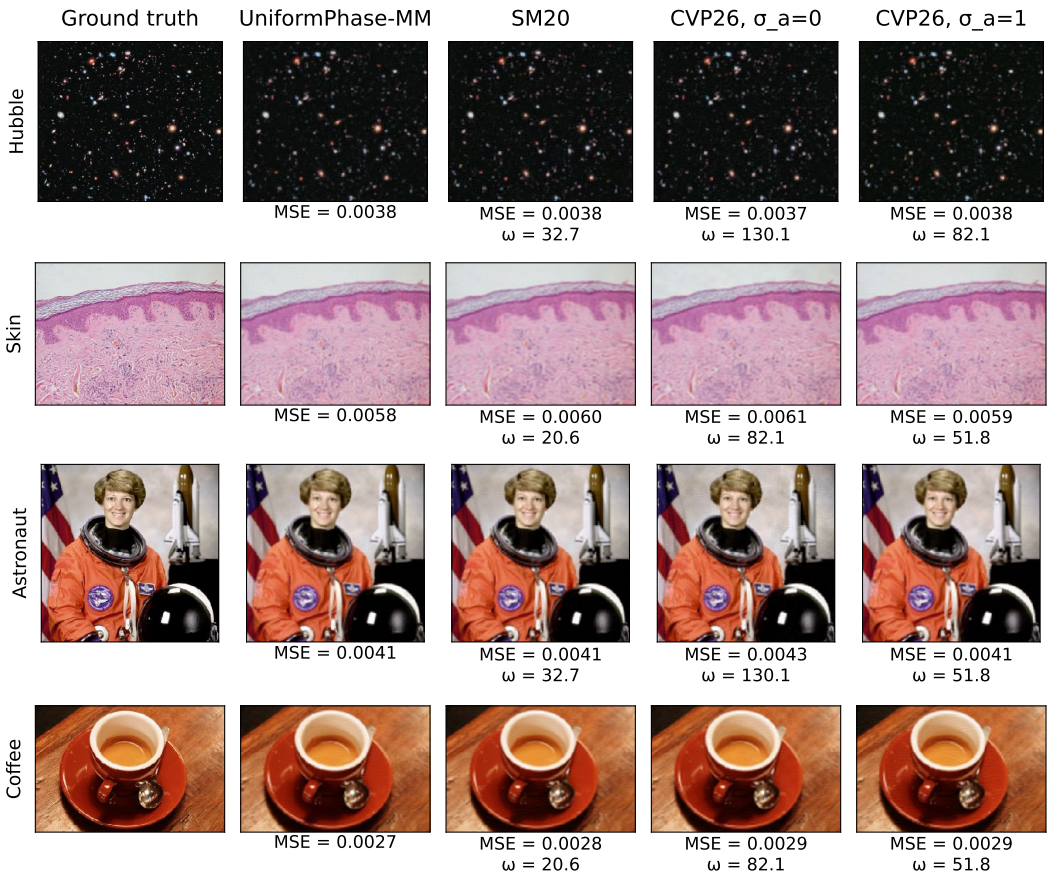

Figure 8: RGB best-ω_0 visual comparisons: Hubble, Skin, Astronaut, and Coffee. Same layout as figure 7.

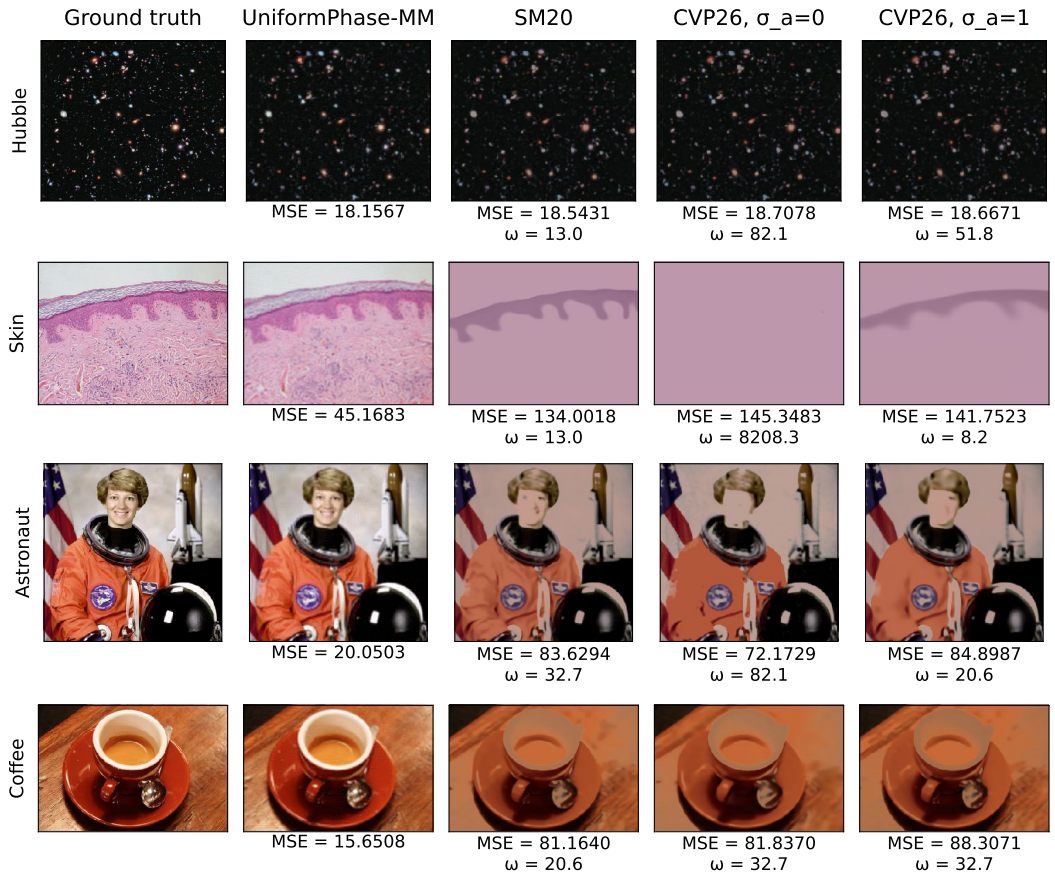

Figure 9: CIELAB best-ω_0 visual comparisons: Hubble, Skin, Astronaut, and Coffee. Predictions are converted from CIELAB to RGB for display; the reported MSE values are computed in CIELAB coordinates. Same layout as figure 7.