Supplementary Materials

A Experimental Details

A.1 Analog Diffusion models (ADMs) on two-dimensional (2D) tasks

For the generation of the 2D distributions reported in Fig. 2, we use a four-layer fully connected model with input and output dimension 2 and hidden dimensions $[15, 16, 13]$. All models are initialized using Xavier initialization [58], after which weights are rescaled by a factor of 71.19. This factor is derived as described in *Supplementary Section C.4*, where the scaling parameters $s_1 = 0.0028$, $s_2 = 4.993$, and $s_3 = 0.0$ are fitted to the AOC activation function.

Models are trained using AdamW with weight decay 10^{-4}, a learning rate of 5×10^{-3}, and a cosine learning-rate schedule with linear warm-up over the first 500 steps. Training is performed for 20,000 steps with a batch size of 256 on 50,000 synthetically generated data points. To construct pairs of data and noise samples, we use the exact optimal transport solver from the POT Python package. Timesteps are sampled uniformly from $[0, 1]$, and no loss reweighting is applied.

A.2 Latent ADMs

For the results shown in Fig. 3, we employ a latent diffusion approach [48] in which generative modeling is performed in the latent space of a pretrained variational autoencoder (VAE), while diffusion inference is carried out using an analog diffusion model. The VAE is trained digitally, and its parameters are frozen prior to diffusion training.

In the first stage, the VAE is trained to reconstruct images from a low-dimensional latent representation by minimizing

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{z \sim q_\phi(z|x)}\big[\log p_\theta(x|z)\big] + \epsilon\, \text{KL}(q_\phi(z|x) \,\|\, p(z))\,, \tag{A1}$$

where $p(z) = \mathcal{N}(0, I)$ denotes a standard Gaussian prior, $q_\phi(z|x)$ is a Gaussian posterior with diagonal covariance parameterized by an encoder network g_ϕ, and $p_\theta(x|z)$ is a Gaussian likelihood parameterized by a decoder network f_θ. We use a small regularization weight $\epsilon = 10^{-6}$ and do not include perceptual or adversarial loss terms. The latent dimensionality is set to $\dim(z) = 10$, which provides reasonable reconstruction fidelity.

The encoder and decoder are convolutional ResNet architectures with three resolution levels (32, 64, and 128 channels), each consisting of two residual blocks. Each block comprises three 3×3 convolutional layers with batch normalization and SiLU activations. Downsampling and upsampling are implemented via average pooling and nearest-neighbor interpolation, respectively. The VAE is trained using Adam with a batch size of 128, a learning rate of 10^{-3}, and no weight decay. All latent diffusion experiments use identical training hyperparameters across datasets.

For the latent ADM, we use a four-layer model with input and output dimension 10 and hidden dimensions $[12, 13, 13]$. Initialization and training follow the same protocol as in the 2D experiments described in *Supplementary Section A.1*.

A.3 ADMs at Scale

All experiments use a latent approach based on the Stable Diffusion VAE (Hugging Face implementation with tag `stabilityai/sd-vae-ft-mse`) which is kept frozen and is not fine-tuned for the specific tasks. We train all models using the standard flow matching formulation with optimal transport path[51]. No loss weighting or rectification is applied. All models are trained with AdamW for 300,000 steps with a batch size of 64, learning rate 10^{-4} and weight decay 10^{-2}. All implicit methods use plain fixed-point iterations to compute fixed-points. More advanced solvers, like Broyden's method can marginally improve the results for the smallest numbers of diffusion steps but are omitted here for simplicity. The fixed-point iteration is initialized at the explicit

Euler update which is taken into account as 1 fixed-point iteration, in particular for the energy calculations.

Digital Twin (DT) & Butterflies

The experiment uses the Smithsonian Butterfly dataset at a resolution of 256×256, with horizontal flips as data augmentation. The digital twin of our hardware is restricted to (i) feedforward (MLP) layers, (ii) tanh activations, and (iii) skip connections. To satisfy these constraints, we adopt the MLP-Mixer architecture[59], which, analogously to transformers, processes images via patchification followed by alternating MLPs over the patch and channel dimensions. To remain as faithful as possible to the hardware constraints, all normalization layers (e.g., LayerNorm) are removed. We employ sinusoidal time embeddings and learned positional encodings. The model consists of 24 blocks. Each block comprises a two-layer MLP with hidden dimension 768 and skip connections for both the patch- and channel-mixing stages respectively, resulting in a total of 58,086,208 parameters. Due to the small size of the dataset (1000 samples) we train on the whole dataset and evaluate FID and KID using the train set as common practice in generative modeling.

Diffusion Transformer & AFHQ

We use the AFHQv2 dataset, comprising the categories cats, dogs, and wild animals, at resolution 256×256 with horizontal flips as data augmentation. We employ a standard Diffusion Transformer (DiT) implementation, specifically DiT-S/2, consisting of 12 transformer blocks with hidden dimension 384, 6 attention heads, and a patch size of 2. The total number of model parameters is 32,572,816. Metrics are computed on a held-out test set comprising 20% of the total data.

U-Net & Oxford Flowers-102

We use the Oxford Flowers-102 dataset, square-cropped and resized to a resolution of 256×256. We adopt the standard U-Net architecture as implemented in the Hugging Face `UNet2DModel`, with 2 ResNet layers per block, 4 downsampling and upsampling stages, and output channel dimensions $[128, 256, 512, 512]$. Cross-attention is applied at the two lowest resolution levels. The total number of model parameters is 106,382,340. Metrics are computed on a held-out test set comprising 20% of the total data.

FLUX.1

We use the FLUX.1 model (Huggingface version with tag `black-forest-labs/FLUX.1-schnell`) and override its sampler to implement implicit solvers. For our qualitative comparison we generate images of 512×512 resolution using different prompts and evaluate the default sampler (explicit Euler) as well as the Crank-Nicolson solver (Implicit-CN) using a maximum of 4 fixed-point iterations. Different to all other experiments, which use simple fixed-point iterations, here we employ a Broyden solver to compute fixed-points as due to the extremely low number of diffusion steps, simple fixed-point iterations became unstable even for Implicit-CN.

B Hardware implementation and characterization

B.1 Hardware implementation

A photograph of the analog optical hardware, used to generate all the experimental datasets is shown in Extended Data Fig. 1d, with key components highlighted in Extended Data Fig. 1b. A schematic diagram of the optical subsystem is provided in Extended Data Fig. 1c. The system comprises a microLED source array, two SLMs, two PD arrays, and an imaging system. Its operating principle builds on our prior platform[20]: light emitted by the sources is collected by an objective lens and imaged onto the SLMs through a combined 4F system formed by the objective and lens groups. The present implementation introduces three key modifications. First, the spreading lens is replaced with a Thorlabs LJ1567L1-A to accommodate the approximately ten-fold increasee in the number of weights. Second, the optical paths corresponding to positive and negative weights are spatially separated rather than interleaved, to reduce optical crosstalk between the two paths. Third, a cylindrical merging lens (Thorlabs LJ1638L1-A) is placed in front of each PD array. This lens compresses the vertical extent of the SLM image on the detector while preserving its horizontal magnification. The image of the reflective SLM is then out of focus in the vertical direction, but this is not a problem as we only require summation of these signals. The normalized response of one SLM as a function of gray value is reported on the top of Supplementary Fig. 1a; both SLMs exhibit comparable performance. The microLED array is custom-fabricated and consists of 48 independently addressable channels, each comprising four green emitters arranged in a 4×48 layout. The emitters are structurally identical to display-grade microLEDs and are operated in parallel within each channel to distribute optical power. Images of the array under an inspection microscope and projected onto a camera are shown in Extended Data Fig. 1b and Supplementary Fig. 1c. Camera images at other points in the system are shown in Supplementary Fig. 1d and Supplementary Fig. 1e. The microLED pitch at the SLM is 193 µm and the SLM plane has 8.5µm pixel pitch and a fill factor of 92%. The end-to-end bandwidth of the optical subsystem, measured with the microLEDs driven at 10 mA, is reported in Supplementary Fig. 1b, using a Keysight Technologies E5061B network analyser. The tested signal was injected at the LED driver board, propagated through the full optical subsystem, and detected at the PD end. The PD output was routed to a breakout board for the measurement. The -3 dB cutoff point occurs at 5 MHz, which reflects the combined response of the microLEDs, PDs, and associated electronic circuitry, and sets the upper bound for analogue iteration speed in the current hardware configuration. The custom electronic control circuitry, implemented using off-the-shelf surface-mount components on printed circuit boards, is described in detail in Supplementary Material B.2 of [20].

In the current prototype, the inter-step conditioning operation (a per-channel affine transform) is performed digitally between diffusion steps and applied as voltages injected at the subsequent diffusion step (corresponding to the additive conditioning term $b + te + pad(x)$ in (5); see Fig. 1b for a schematic). All iterative denoising updates within each diffusion step are executed in the analog domain. Fully analog state propagation across diffusion steps would require analog memory together with programmable gain and offset. Such analog memory can be implemented using a sample-and-hold circuit.

B.2 DT vs hardware accuracy

To quantify the accuracy of the hardware to its DT, we perform four complementary calibration and validation experiments. First, we assess open-loop accuracy by measuring the output of the matrix multiplication path alone across a range of gain values β (Supplementary Fig. 2a). For each β, we test 100 randomly sampled weight matrices and 6 representative input voltages, spanning the dynamic range of the analog tanh nonlinearity, for a total of 600 experiments per gain setting. In each case, hardware outputs are compared to the DT predictions, and the normalized root-mean-square error (NRMSE) is averaged over 100 repeated measurements. Across all 4200 experiments, we observe a mean NRMSE of 0.015. An example error distribution for $\beta = 30$ is shown in the inset of Supplementary Fig. 2a.

Second, we assess closed-loop accuracy through 1,000 experiments spanning 100 randomly sampled weight matrices, 20 channel-wise gain values α, and per-channel β settings randomly chosen to match the operating range used in diffusion experiments. Hardware outputs are compared against their DT predictions in Supplementary

Fig. 2b, revealing high linearity and close agreement across the full voltage range, with a mean NRMSE of 0.009.

To quantify accuracy during diffusion inference, we measure the per-step NRMSE for each diffusion step by comparing the hardware output of a single denoising update against its corresponding DT evaluation (Supplementary Fig. 2c for 2D tasks, and Supplementary Fig. 2e for latent-space tasks). Aggregating results across multiple 2D diffusion tasks (Supplementary Fig. 1c) yields a mean per-step NRMSE of 0.013, while across multiple latent-space experiments (Supplementary Fig. 3e) achieve a mean NRMSE of 0.01.

Because the diffusion model is trained under the assumption of ideal correspondence between analog and digital execution small per-step discrepancies accumulate along the sampling trajectory. We therefore also report the accumulated NRMSE relative to the DT as a function of diffusion step index (Supplementary Fig. 2d and Supplementary Fig. 2f. Although the accumulated error increases with depth, it remains within a range compatible with successful generation across all 2D and latent benchmarks.

Finally, we observe systematic differences between the error distributions in 2D and latent experiments. This likely reflects the general calibration strategy of the DT, whose parameters are fitted using random weight matrices rather than optimized for task-specific operating regimes, promoting generalization across workloads.

B.3 Noise

Noise in the hardware arises from LED relative-intensity noise, photodetector shot and thermal noise, electronic dark noise, and temporal fluctuations in the optical weights. Among these, SLM phase flicker dominates, causing time-varying perturbations to the effective weight matrix. Supplementary Fig. 1a shows the measured output-voltage distributions across grey levels in an open loop experiment. Since this measurement is taken at the end of the optical subsystem, primarily driven by the SLM flicker, and represents the effective noise injected into the computation. We then evaluate trial-wise averaging, which reduces uncorrelated SLM-induced noise. Extended Data Fig. 7a shows that averaging over five repeats improves the per-step NRMSE from $4.1{\times}10^{-2}$ to $1.9{\times}10^{-2}$ for 2D experiments, and negaligible improvment beyond 5 averages for all datasets (Extended Data Fig. 7a and Extended Data Fig. 8c-d). Importantly, this averaging cost is not fundamental: if a measurement is synchronised to the SLM flicker, its temporal waveform could be incorporated into the DT, potentially eliminating the need for experimental averaging.

C ADMs: Additional Technical Details

This section provides technical results and implementation details that complement the main-text Methods. We refer to Methods for definitions and notation. Here we provide: (i) a contraction-based convergence guarantee and proof for the fixed-point inference of ADMs, (ii) a finite-iteration convergence statement for explicit diffusion models on AOC, (iii) details on α-relaxed fixed-point iterations and how they enforce optical gain constraints (including channel-wise relaxation), and (iv) the procedure to compile standard neural network models to the AOC DT.

C.1 Convergence of implicit fixed-point inference

Below we will make use of the following simple result.

Lemma 2 *Let $f : \mathbb{R}^n \to \mathbb{R}^m$ denote a K-layer neural network defined via composition as*

$$f(x) = (L_K \circ \ldots \circ L_1)(x), \quad L_i(x) = W_i\sigma(x) + b_i, \ i = 1, \ldots, K. \tag{C2}$$

Then the Lipschitz constant $\mathrm{Lip}(f)$ *of f is given as*

$$\mathrm{Lip}(f) = \mathrm{Lip}(\sigma)^K \prod_{i=1}^{K} \|W_i\|_\infty. \tag{C3}$$

Proof The Lipschitz constant of every layer L_i can be estimated via

$$\|L_i(x) - L_i(y)\|_\infty = \|W_i\sigma(x) - W_i\sigma(y)\|_\infty \leq \|W_i\|_\infty \|\sigma(x) - \sigma(y)\|_\infty \tag{C4}$$

$$\leq \|W_i\|_\infty \operatorname{Lip}(\sigma)\|x - y\|_\infty, \tag{C5}$$

to be $\operatorname{Lip}(f) = \operatorname{Lip}(\sigma)\|W_i\|_\infty$. With that the result follows recursively. □

Consider the ADM operator $\mathrm{ADM}_{x,t;\theta(\Delta t)}$ defined in Methods (Eqs. (5) and (6)). The following lemma provides a sufficient condition for existence and uniqueness of a fixed point.

Lemma 3 (Fixed-point existence and convergence) *Let σ be $\operatorname{Lip}(\sigma)$-Lipschitz and let $\|\cdot\|_\infty$ denote the induced matrix norm. If $\Delta t < \frac{1}{\operatorname{Lip}(\sigma)^K \|W\|_\infty^K}$, then for any (x, t) the operator $\mathrm{ADM}_{x,t;\theta(\Delta t)}$ has a unique fixed point $z^* \in \mathbb{R}^n$ and, for any initialization z_0, we have $\lim_{k\to\infty} \mathrm{ADM}^k_{x,t;\theta(\Delta t)}(z_0) = z^*$.*

Proof Consider the ADM operator (6) with $L_i(z_i) = W_i\sigma(z_i) + b_i + te_i + [\mathrm{pad}(x)]_{d_i:d_{i+1}}$ and $L_K^{\Delta t}(z_K) = \Delta t L_K(z_K)$. Consider any $z \in \mathbb{R}^n$ with structure $z = (z_1, \ldots, z_K)^T$, then after K applications of the ADM operator we obtain

$$\left(\mathrm{ADM}_{x,t;\theta}\right)^K(z) = \begin{pmatrix} L_K^{\Delta t} \circ \ldots \circ L_1(z_1) \\ L_1 \circ L_K^{\Delta t} \circ \ldots \circ L_2(z_2) \\ L_2 \circ L_1 \circ L_K^{\Delta t} \circ \ldots \circ L_3(z_3) \\ \vdots \\ L_{K-1} \circ L_{K-2} \circ \ldots \circ L_1 \circ L_K^{\Delta t}(z_K) \end{pmatrix} =: \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \\ F_K \end{pmatrix}(z), \tag{C6}$$

where each entry applies a permutation of the layers $\{L_1, \ldots, L_K^{\Delta t}\}$ of the network. By Lemma 2 we thus have $\operatorname{Lip}(F_j) = \operatorname{Lip}(\sigma)^K \Delta t \prod_{i=1}^K \|W_i\|_\infty$ for $j = 1, \ldots, K$. Note that the presence of Δt is due to the scaling of the last layer in (6). Consider any $u, v \in \mathbb{R}^n$,

$$\left\|\left(\mathrm{ADM}_{x,t;\theta_{\Delta t}}\right)^K(u) - \left(\mathrm{ADM}_{x,t;\theta_{\Delta t}}\right)^K(v)\right\|_\infty \tag{C7}$$

$$\leq \left\|\left(\mathrm{ADM}_{x,t;\theta_{\Delta t}}\right)^K\right\|_{\mathrm{op}} \|u - v\|_\infty \tag{C8}$$

$$= \left(\max_i \mathrm{Lip}(F_i) \right) \|u - v\|_\infty \tag{C9}$$

$$\leq \mathrm{Lip}(\sigma)^K \Delta t \left(\prod_{i=1}^{K} \|W_i\|_\infty \right) \|u - v\|_\infty \tag{C10}$$

$$\leq \underbrace{\mathrm{Lip}(\sigma)^K \Delta t \|W\|_\infty^K}_{=\mathrm{Lip}(\mathrm{ADM}_{x,t;\theta_{\Delta t}})} \|u - v\|_\infty \tag{C11}$$

Now with a recursion argument we have $\mathrm{Lip}\left(\mathrm{ADM}_{x,t;\theta_{\Delta t}}\right) = \mathrm{Lip}\left(\left(\mathrm{ADM}_{x,t;\theta_{\Delta t}}\right)^K\right)^{1/K} = (\Delta t)^{1/K}\,\mathrm{Lip}(\sigma)\|W\|_\infty$. Thus for $\Delta t < \frac{1}{\mathrm{Lip}(\sigma)^K \|W\|_\infty^K}$, the ADM operator becomes a contraction, i.e., $\mathrm{Lip}\left(\mathrm{ADM}_{x,t,\theta_{\Delta t}}\right) < 1$, and the result follows by Banach's fixed-point theorem. □

Given that a unique fixed-point exists the following result shows that it can be identified with the implicit Euler update.

Theorem 4 (Implicit ADM) *Under the conditions of Lemma 3, the readout components $[z^*]_{:d}$ of the unique fixed point z^* of the ADM operator solve the implicit Euler equation, i.e.,*

$$[z^*]_{:d} = x_t + \Delta t\, f_\theta([z^*]_{:d}, t + \Delta t). \tag{C12}$$

Proof As the conditions of Lemma 3 are fulfilled, there exists $z^* \in \mathbb{R}^n$ such that $\mathrm{ADM}_{x_t,t;\theta_{\Delta t}}(z^*) = z^*$ and consequently $\left(\mathrm{ADM}_{x_t,t;\theta_{\Delta t}}\right)^K(z^*) = z^*$. Applying the ADM operator K times we obtain

$$\left(\mathrm{ADM}_{x_t,t;\theta_{\Delta t}}\right)^K(z^*) = \begin{pmatrix} L_K^{\Delta t} \circ \ldots \circ L_1(z_1^*) \\ L_1 \circ L_K^{\Delta t} \circ \ldots \circ L_2(z_2^*) \\ L_2 \circ L_1 \circ L_K^{\Delta t} \circ \ldots \circ L_3(z_3^*) \\ \vdots \\ L_{K-1} \circ L_{K-2} \circ \ldots \circ L_1 \circ L_K^{\Delta t}(z_K^*) \end{pmatrix} = \begin{pmatrix} \mathrm{Euler}_{x_t,\Delta t}(z_1^*) \\ z_2^* \\ z_3^* \\ \vdots \\ z_K^* \end{pmatrix} = \begin{pmatrix} z_1^* \\ z_2^* \\ z_3^* \\ \vdots \\ z_K^* \end{pmatrix} \tag{C13}$$

where we defined $\mathrm{Euler}_{x_t,\Delta t}(z) := x_t + f(z)\Delta t$. Now the first blockentry reads $\mathrm{Euler}_{x_t,\Delta t}(z_1^*) = z_1^*$, which identifies $z_1^* = [z^*]_{:d}$ with the implicit Euler update $x_{t+\Delta t}$

as defined in (4). Now with z^* being a fixed-point of the ADM operator, we have $\left[\left(\mathrm{ADM}_{x_t,t;\theta_{\Delta t}}\right)^k (z^*)\right]_{:d} = [z^*]_{:d} = x_{t+\Delta t}$ for any $k \in \mathbb{N}_0$. $\square$

Theorem 1 follows directly by combining Lemma 3 with Theorem 4.

C.2 Explicit Models on analog hardware

In addition to ADMs, which implement an implicit ODE solver at inference time, we provide a formulation of conventional explicit ODE solvers which is immediately compatible with AOC and fixed-point search. In the following we refer to these models as explicit models. Explicit models use an alternative block structure in which a single forward pass through all layers is completed after exactly $K+1$ AOC iterations. In this case, the fixed-point coincides with the explicit network evaluation, and the last d components of the AOC state yield the output. Explicit models are obtained by placing the matrices corresponding to the layers on the off-diagonal. Analogous to the implicit version we define the parameters of the explicit model as $\theta^e(\Delta t) = \{W^e_{\Delta t}, b^e_{\Delta t}, e^e_{\Delta t}\}$, with

$$W^e_{\Delta t} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0_{d\times d} \\ W_1 & 0 & 0 & \cdots & 0 \\ 0 & W_2 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & \Delta t\, W_K & 0 \end{bmatrix}, \quad b^e_{\Delta t} = \begin{bmatrix} 0 \\ b_1 \\ b_2 \\ \vdots \\ \Delta t\, b_K \end{bmatrix}, \quad e^e_{\Delta t} = \begin{bmatrix} 0 \\ e_1 \\ e_2 \\ \vdots \\ \Delta t\, e_K \end{bmatrix}. \tag{C14}$$

Evaluation of the model at training time differs from ADMs only in the readout components,

$$f_\theta(x,t) = \left[\left(\mathrm{ADM}_{x=0,t;\theta^e(\Delta t=1)}\right)^K (x)\right]_{-d:}, \tag{C15}$$

where $[\cdot]_{-d:}$ selects the last d components. Using an input embedding that injects the current iterate into both the top and readout blocks $\mathrm{pad}^e(x) = (x, 0, \ldots, 0, x)^\top \in \mathbb{R}^n$, the explicit Euler update is recovered from the last d components of the fixed point of

$\mathrm{ADM}_{x,t;\theta^e(\Delta t)}(z) = W^e_{\Delta t}\sigma(z) + b^e_{\Delta t} + te^e_{\Delta t} + \mathrm{pad}^e(x)$, i.e. $[z^*]_{-d:}$, which is found after exactly $K+1$ iterations, which is formalized next.

Proposition 5 (Explicit Models) *Let $z^{(k+1)} = \mathrm{ADM}_{x_t,t;\theta^e(\Delta t)}(z^{(k)})$ denote the explicit ADM iteration. Then for any initialization $z^{(0)}$ and any $K' \geq K+1$ we have $z^{(K')} = z^{(K+1)}$, and the readout satisfies*

$$[z^{(K')}]_{-d:} = x_t + \Delta t\, f_\theta(x_t, t), \tag{C16}$$

i.e. the explicit Euler update.

Proof By construction, each application propagates the signal one layer forward without feedback to the top block. After $K+1$ iterations, all blocks depend only on x_t, and a further application does not change the state. The last block equals $x_t + \Delta t f_\theta(x_t, t)$ due to the injected copy of x_t in the last d components of $\mathrm{pad}^e(x_t)$. $\square$

While explicit models reach a stationary state similar to ADMs, they lack the attractor dynamics of the implicit formulation: after $K+1$ iterations the result gets continuously re-computed with different realizations of the zero-mean analog noise, which at first glance allows for averaging as in the implicit case. However, as the analog noise undergoes non-linear transformations during the K+1 iterations needed to reach the readout components, the total corruption is not necessarily centered at zero anymore, which limits the usefulness of averaging. In contrast, implicit models are self-correcting as each individual application of the AOC operator can be interpreted as a gradient descent step towards the fixed-point, corrupted by zero-mean analog noise. This corresponds to stochastic gradient descent, which has well-explored convergence properties.

C.3 Relaxed fixed-point iterations

The optical vector–matrix multiplication (OVMM) unit in the AOC represents weights using SLMs, which natively encode values in a bounded range. Using paired modulators for positive and negative weights allows matrices with entries in $[-1, 1]$ to be represented. General weight matrices $W \in \mathbb{R}^{n\times n}$ are therefore implemented by normalizing to unit range and applying an optical amplification factor $\beta = \|W\|_\infty$.

In practice, additional signal loss arises from the optical fan-out of light across SLM pixels, particularly for sparse or highly unbalanced matrices such as the identity, where most pixels block light. These effects require further amplification to maintain sufficient signal strength. However, large amplification degrades the signal-to-noise ratio, imposing a practical upper bound $\beta \leq \beta_{\max}$. To allow representing arbitrary models, this motivates the relaxed fixed-point formulation introduced in Methods, which preserves the fixed-point of the model while reducing the required optical gain,

$$\begin{aligned} \mathrm{ADM}^{\alpha}_{x_t,t;\theta}(z) &:= \alpha \odot z + (1-\alpha) \odot \mathrm{ADM}_{x_t,t;\theta}(z) \\ &= \alpha \odot z + W_\alpha \sigma(z) + b_\alpha + t e_\alpha + x_t^\alpha, \end{aligned} \tag{C17}$$

where $\alpha \in [0, 1)^n$ is fixed and $W_\alpha = (I - \mathrm{diag}(\alpha))W$, $b_\alpha = (1-\alpha) \odot b$, $e_\alpha = (1-\alpha) \odot e$, $x_t^\alpha = (1-\alpha) \odot x_t$. This relaxation reduces the required effective OVMM gain without modifying the solution of the fixed-point equation.

Theorem 6 *For any $\alpha \in \mathbb{R}^n$ with $\alpha_i \in [0, 1)$, $i = 1, \ldots, n$, the α-relaxed ADM operator converges under the same step-size conditions as the standard ADM operator. Moreover, both operators have exactly the same fixed points.*

Proof If $z^* = \mathrm{ADM}_{x_t,t;\theta}(z^*)$, then $\mathrm{ADM}^{\alpha}_{x_t,t;\theta}(z^*) = \alpha \odot z^* + (1-\alpha) \odot z^* = z^*$. Conversely, if $z^* = \mathrm{ADM}^{\alpha}_{x_t,t;\theta}(z^*)$, then $(1-\alpha) \odot z^* = (1-\alpha) \odot \mathrm{ADM}_{x_t,t;\theta}(z^*)$, and since $1-\alpha$ has strictly positive entries, this implies $z^* = \mathrm{ADM}_{x_t,t;\theta}(z^*)$. Contractivity follows as ADM^{α} is a convex combination of the identity and $\mathrm{ADM}_{x_t,t;\theta}$, and thus satisfies the same contraction condition as the unrelaxed operator. □

For a vector $z \in \mathbb{R}^n$ let $\mathrm{diag}(z) \in \mathbb{R}^{n\times n}$ denote the square matrix with z on the diagonal. From $W_\alpha = (I - \mathrm{diag}(\alpha))W$, it follows that $\|W_\alpha\|_\infty \leq (1-\alpha_{\min})\|W\|_\infty$. Hence, for a given $\beta_{\max}$, any matrix W with $\beta = \|W\|_\infty$ can be implemented by choosing $1 - \beta_{\max}/\beta \leq \alpha_i < 1$. Using a uniform relaxation $\alpha_i(\beta) = 1 - \beta_{\max}/\beta$ yields $W_\alpha = \beta_{\max} W_{\mathrm{norm}}$ and the relaxed operator

$$\mathrm{ADM}^{\alpha(\beta)}_{x_t,t;\theta}(z) = \alpha(\beta) \odot z + \beta_{\max} W_{\mathrm{norm}} \sigma(z) + b_{\alpha(\beta)} + t e_{\alpha(\beta)} + x_t^{\alpha(\beta)}.$$

To further reduce optical power loss, we normalize W row-wise, yielding $W = \beta\, \mathrm{diag}(a)\, W_{\mathrm{norm,cw}}$. The channel-wise amplification factors a_i can be absorbed into a channel-wise relaxation by choosing $1 - \alpha_i^{\mathrm{cw}} = 1/a_i$. Imposing the gain constraint $\beta_{\max}$ gives $\alpha_i^{\mathrm{cw}}(\beta) = 1 - \beta_{\max}/(a_i\beta)$ and the operator becomes

$$\mathrm{ADM}^{\alpha^{\mathrm{cw}}(\beta)}_{x_t,t;\theta}(z) = \alpha^{\mathrm{cw}}(\beta) \odot z + \beta_{\max} W_{\mathrm{norm,cw}} \sigma(z) + b_{\alpha^{\mathrm{cw}}(\beta)} + t e_{\alpha^{\mathrm{cw}}(\beta)} + x_t^{\alpha^{\mathrm{cw}}(\beta)}.$$

This channel-wise relaxation substantially reduces OVMM power loss and improves the effective signal-to-noise ratio while preserving the ADM fixed point at the expense of slower fixed-point convergence.

C.4 Compiling neural networks to the AOC DT

The DT compiles neural network models expressed in a conventional sequential form into an AOC-compatible block-matrix representation (see Methods). While the DT can, in principle, represent the same computations as its digital counterpart, the activation function implemented in hardware differs from the standard hyperbolic tangent by input and output scalings and an additive offset. Since tanh is not scale invariant, these differences must be compensated to preserve the behavior of standard initializations.

The AOC implements the iterative update

$$\mathrm{ADM}_{x,t;\theta}(z) = W\,\sigma_{\mathrm{AOC}}(z) + b + te + x, \tag{C18}$$

where $\sigma_{\mathrm{AOC}}(u) = s_1\sigma(s_2 u) + s_3$ is a scaled and shifted tanh. A corresponding conventional digital neural network with tanh activations implements

$$f_{x,t;\theta}(z) = W\,\sigma(z) + b + te + x. \tag{C19}$$

To leverage standard initialization and scaling heuristics developed for networks of the form (C19)[58], we introduce scaling factors s_1, s_2 and s_3 to match the fixed-points of AOC and the conventional neural net.

Specifically, equivalence of the fixed points of (C18) and (C19) is ensured by the parameter rescaling

$$W_{\mathrm{AOC}} = \frac{1}{s_1 s_2}W, \qquad b_{\mathrm{AOC}} = \frac{1}{s_2}b - s_3, \qquad e_{\mathrm{AOC}} = \frac{1}{s_2}e, \qquad x_{\mathrm{AOC}} = \frac{1}{s_2}x, \tag{C20}$$

with the corresponding output relation $z^* = s_2 z^*_{\mathrm{AOC}}$. Under this transformation, the DT fixed-point equation reduces exactly to the standard tanh fixed-point equation. As the output of each diffusion step serves as the input to the next, no intermediate rescaling is required between diffusion steps; scaling is applied only once, before the first and after the final diffusion step.

D Evaluation metrics

D.1 Normalized root mean squared error (NRMSE)

The NRMSE measures point-wise deviation between evaluated values z_k and reference values z_k^{ref} over N samples:

$$\mathrm{NRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}\|z_k - z_k^{\mathrm{ref}}\|_2^2}}{\sqrt{\frac{1}{N}\sum_{k=1}^{N}\|z_k^{\mathrm{ref}}\|_2^2}}. \tag{D21}$$

D.2 Maximum mean discrepancy (MMD)

MMD measures the distance between the distributions of evaluated values $\{z_k\}_{k=1}^{N}$ and reference values $\{z_l^{ref}\}_{l=1}^{M}$ over N evaluated values and M reference values by measuring the distance of their kernel mean embeddings in a reproducing kernel Hilbert space. We use a standard Gaussian kernel $\kappa(\mathbf{z}, \mathbf{z}')$.

$$\mathrm{MMD}^2 = \frac{1}{N(N-1)}\sum_{k\neq k'}\kappa(z_k, z_{k'}) + \frac{1}{M(M-1)}\sum_{\ell\neq\ell'}\kappa(z_\ell^{\mathrm{ref}}, z_{\ell'}^{\mathrm{ref}}) - \frac{2}{NM}\sum_{k=1}^{N}\sum_{\ell=1}^{M}\kappa(z_k, z_\ell^{\mathrm{ref}}). \tag{D22}$$

We refer to MMD^2 as MMD in the text.

D.3 Wasserstein-2 (W2) distance

For probability distributions μ and ν on a metric space, the Wasserstein-2 distance $W_2(\mu, \nu)$ measures the minimal expected squared cost of transporting mass from μ to ν:

$$W_2^2(\mu, \nu) = \inf_{\pi\in\Pi(\mu,\nu)} \int \|x - y\|^2 \,\mathrm{d}\pi(x, y),$$

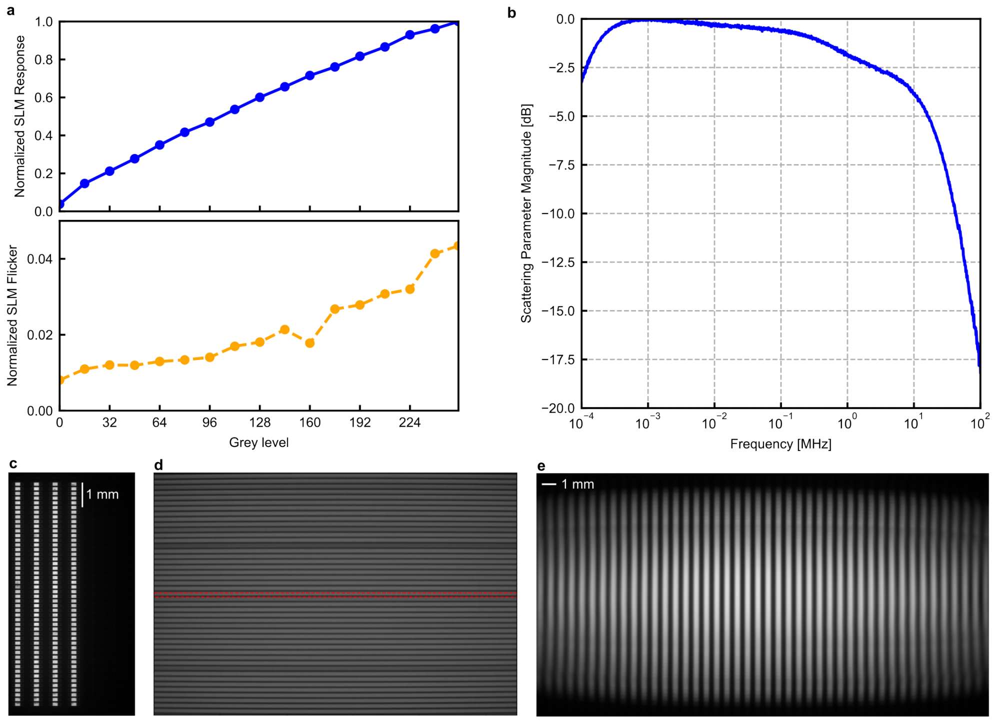
where π ranges over all joint distributions with marginals μ and ν. In practice, W_2^2 is computed by solving a discrete optimal transport linear program between empirical distributions with uniform sample weights and cost matrix $C_{ij} = \|x_i - y_j\|^2$, returning the minimum total transport cost. We use the POT Python package to compute W2.

D.4 Fréchet Inception Distance (FID)

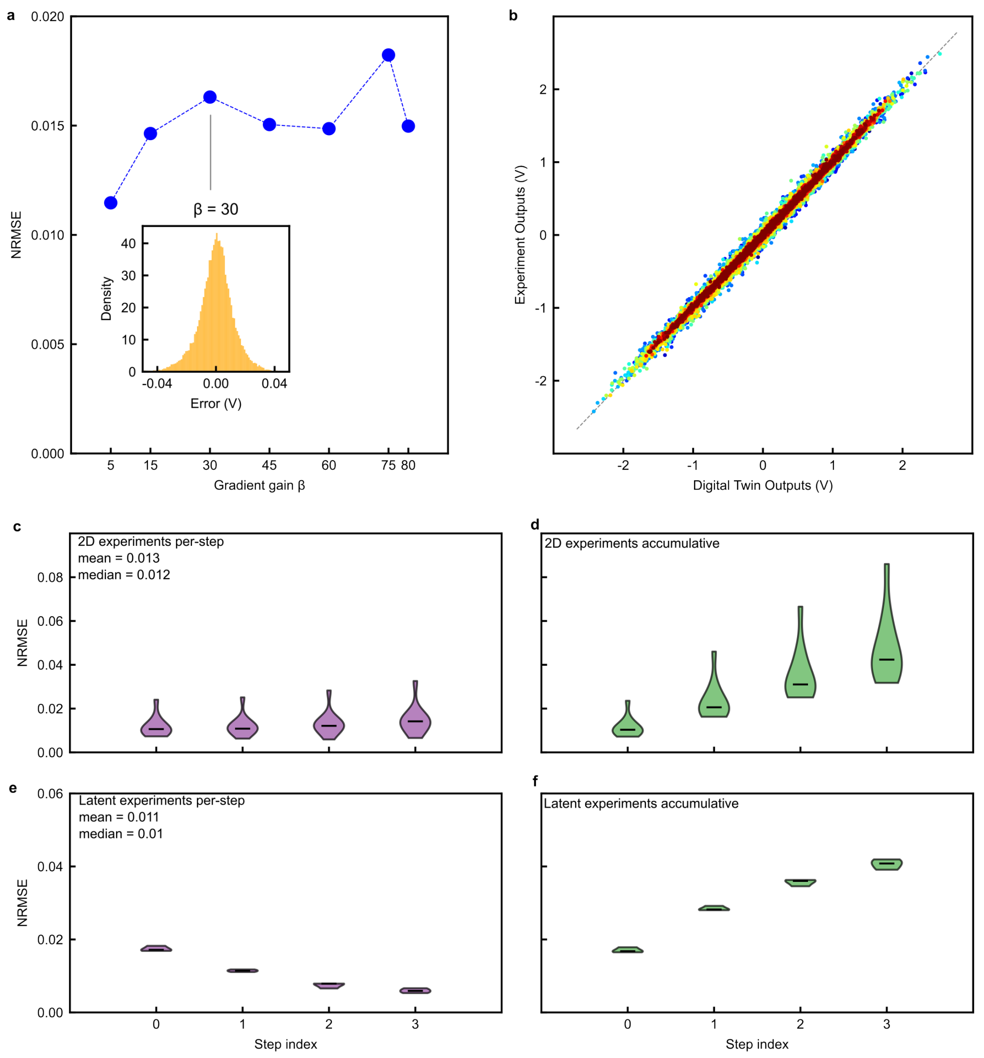
Fréchet Inception Distance (FID) is a distributional similarity metric that quantifies the discrepancy between real and generated image distributions in the feature space of a pretrained Inception-v3 network. Specifically, deep features are extracted from an intermediate layer of Inception-v3 for both real and generated images, and each feature distribution is approximated by a multivariate Gaussian. The FID corresponds to the closed-form 2-Wasserstein (Fréchet) distance between these Gaussian distributions, thereby capturing differences in both the mean (image fidelity) and covariance (sample diversity) of the feature representations. Lower FID values indicate closer alignment between generated and real image distributions.

chngcntr

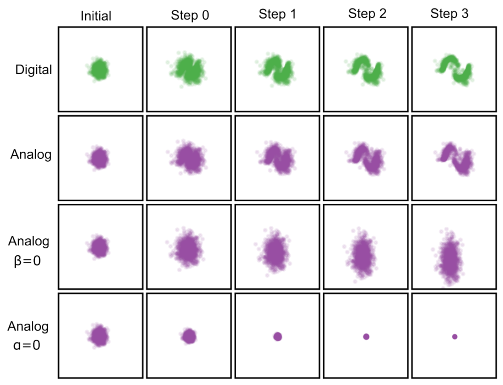
Supplementary Figures



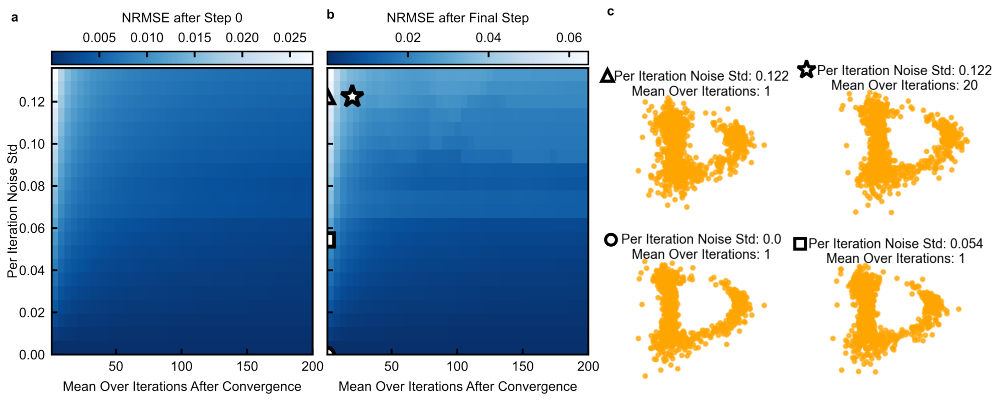
Supplementary Fig. 1: Device and system characterization. **a,** Top: normalized static SLM response versus grey level. Bottom: normalized flicker amplitude measured at the matrix multiplier output; each distribution includes all the noise sources present in the opto-electronic system. Both SLMs present very comparable performances. **b,** Measured end-to-end bandwidth of the optical sub-system. Scattering parameter magnitude (dB) versus frequency for the LED–SLM–PD chain, measured using a Keysight E5061B network analyser. The micro-LEDs are driven at 10 mA, and the PD output was accessed via a breakout board. The -3 dB point occurs at 5 MHz, indicating the effective bandwidth of the current hardware. **c,** Camera image at the SLM plane showing the projected 4x48 LED array. The LEDs (25µm emitting height, 50µm pitch), are magnified by 3.86× and imaged onto the SLM plane, defining the optical input channels used for vector encoding. **d,** Camera at the same plane after inserting the merger lens, which optically combines the 4x48 LED array into 48 horizontal illumination stripes. When the SLM is placed at this plane, each stripe aligns with a row of programmable weights. The SLM has an 8.5µm pixel pitch and a 92% fill factor; each weight element is implemented using a block of 14x16 SLM pixels. **e,** Camera at the PD plane with vertical stripe patterns displayed on the SLM, illustrating how the merger performs optical summation across input channels.



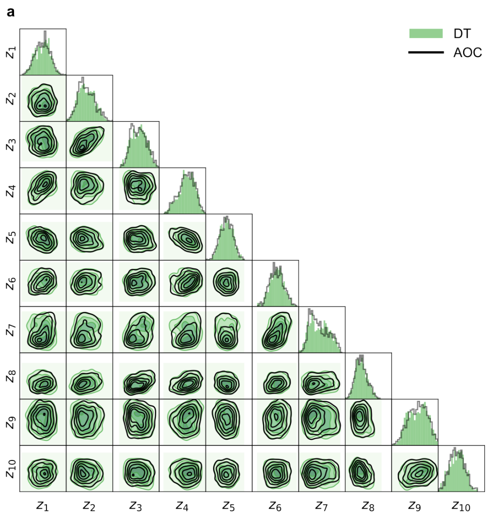
Supplementary Fig. 2: Accuracy characterization of the analog optical hardware against digital twin. **a,** Open-loop accuracy of the matrix-multiplication path across gain settings β, reported as normalized root-mean-square error (NRMSE) averaged over 600 experiments per setting (100 random weight matrices $\times$ 6 input voltages). Inset shows the error distribution for $\beta = 30$. **b,** Closed-loop agreement between analog and digital-twin outputs over 1,000 experiments spanning random weight matrices and channel-wise gains α, β (mean NRMSE $= 0.009$). Colours demote different output channels. **c-f** Per-step and cumulative NRMSE during diffusion inference for 2D distributions (c, d) and latent-space tasks (e,f), aggregated across benchmarks. Horizontal bars indicate the median.

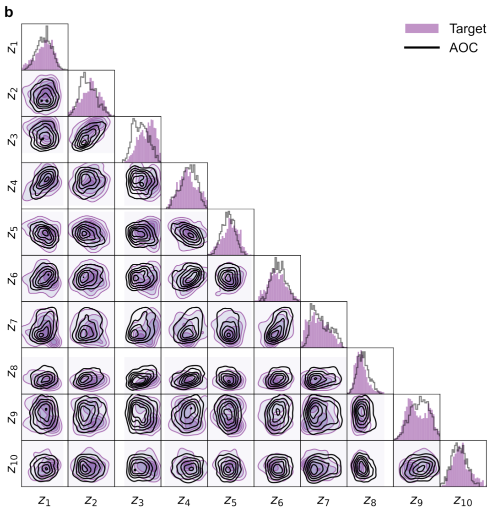


Supplementary Fig. 3: Sensitivity of analog generation to hardware parameter settings to highlight the role of different parts of the system. Rows correspond to different execution modes: digital baseline, analog hardware with nominal parameter settings, analog hardware with optical gain disabled, i.e. optical subsystem off (β=0), and analog hardware with skip-connection disabled (α=0). Here, β denotes the optical amplification applied to the W$\sigma(z)$ term in (5), and α denotes the analog skip-connection fraction defined in (10). Columns show the evolution from the initial noise state through successive diffusion steps. Disabling either β or α prevents convergence to the target distribution, demonstrating that both optical and annealing dynamics are essential for successful analog diffusion inference.



Supplementary Fig. 4: DT noise-tolerance characterization of implicit inference under per-iteration Gaussian noise. **a,** Using an examplar 2D distribution (letter D), we inject additive per-iteration Gaussian noise in the DT and report the NRMSE at diffusion step 0 as a function of standard deviation and number of iterations averaged. The noise standard-deviation range corresponds to levels observed on hardware under typical operating conditions. This quantifies intrinsic noise tolerance of fixed-point search and the benefit of in-loop averaging. **b,** Same analysis as in **a**, but reporting NRMSE at the final diffusion step after completing the full diffusion trajectory, capturing cumulative effect of per-iteration noise across per fixed-point search and across diffusion steps. **c,** Representative 2D distributions generated by the DT for two noise levels (standard deviation = 0 0.054, and 0.122) and two averaging settings (mean over iterations=1 and 20), illustrating that implicit diffusion inference preserves distributional structure under noise and that increased averaging improves sample fidelity. In analog hardware, this averaging incurs low marginal cost.





Supplementary Fig. 5: Latent space distributions comparison for the MNIST experiment, chosen as a representative example. **a,** Corner plot showing marginal (diagonal) and joint (lower triangle) distributions of latent variables z_1-z_{10} for the digital twin baseline (DT, green) and the AOC results (black contours). Diagonal panels show 1D histograms for each latent variable, while off-diagonal panels display kernel density contours representing pairwise joint distributions. For consistency, all off-diagonal axes ranges are limited to [-10, 10] for comparability. **b,** Same representation comparing AOC (black contours) with target distribution (purple).