Analog Diffusion Models

Jiaqi Chu

Jiaqi.Chu@microsoft.com

Microsoft Research https://orcid.org/0009-0008-4744-334X

Heiner Kremer

Microsoft Research

Fabian Falck

Microsoft Research

Grace Brennan

Microsoft research https://orcid.org/0000-0001-7081-0509

Burcu Canakci

Microsoft Research

James Clegg

Microsoft Research Ltd https://orcid.org/0009-0002-3428-7161

Daniel Cletheroe

Microsoft Research (United Kingdom) https://orcid.org/0009-0003-6444-9149

Doug Kelly

Microsoft Research

Christos Gkantsidis

Microsoft Research https://orcid.org/0000-0002-6898-2368

Michael Hansen

Microsoft Research

Paul Jeha

Microsoft Research

Kirill Kalinin

Microsoft Research

Jim Kleewein

Microsoft

Babak Rahmani

Microsoft Research

Saravan Rajmohan

Microsoft

Victor Rühle

Microsoft https://orcid.org/0000-0002-8957-7628

Jannes Gladrow
Microsoft Research
Francesca Parmigiani
Microsoft Research https://orcid.org/0000-0001-7784-2829
Hitesh Ballani
Microsoft Research https://orcid.org/0000-0003-1573-3314

Physical Sciences - Article

Keywords: generative AI, analog computing, diffusion models, flow matching, optical computing, neuromorphic computing, fixed-point compute paradigm

Posted Date: March 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8919479/v1

Additional Declarations: There is **NO** Competing Interest.

Analog Diffusion Models

Heiner Kremer[*,1], Fabian Falck[*,1], Jiaqi Chu[*,1,†], Grace Brennan[1], Burcu Canakci[1], James H. Clegg[1], Daniel Cletheroe[1], Douglas J. Kelly[1], Christos Gkantsidis[1], Michael Hansen[2], Paul Jeha[1], Kirill P. Kalinin[1], Jim Kleewein[2], Babak Rahmani[1], Saravan Rajmohani[2], Victor Rühle[2], Jannes Gladrow[1], Francesca Parmigiani[1,‡], & Hitesh Ballani[1]

[1]Microsoft Research, Cambridge, UK.
[2]Microsoft, Redmond, WA, USA.
[*]These authors contributed equally to this work.
[†]Correspondence to: Jiaqi.Chu@microsoft.com.
[‡]Correspondence to: Francesca.Parmigiani@microsoft.com.

Abstract

As generative artificial intelligence (GenAI) drives computational demands to unprecedented scales[1,2], digital hardware is approaching fundamental limits[3]. Analog and optical systems[4–20] promise orders-of-magnitude efficiency gains, but translating these to application-level gains is challenging due to the mismatch between hardware primitives and algorithmic requirements. Here, we introduce Analog Diffusion Models (ADMs) which implement diffusion inference with an implicit integration scheme, formulating each diffusion step as a fixed-point problem amenable for acceleration by efficient analog hardware. At the same time, training remains identical to that of conventional diffusion models, allowing adoption of established scalable training approaches with no additional overhead. We validate ADMs on analog hardware using three-dimensional optics with 2,304 programmable weights. On hardware, we generate two-dimensional distributions and latent-space distributions for MNIST, FashionMNIST, and ExtendedMNIST, demonstrating the feasibility of executing multi-layer diffusion processes entirely on noisy, non-traditional hardware. The current prototype reaches fixed-point convergence in 10–15 μs per diffusion step, with projections to nanosecond-scale convergence with miniaturization. In simulation, across multiple datasets, backbone architectures, and model sizes ranging from 32 million

to 13 billion parameters, ADMs match the sample quality of standard methods with up to 16× fewer diffusion steps. Most importantly, they could achieve efficiency gains of more than 100× at the application level without sacrificing generation quality, 100× from hardware acceleration, and an additional 1-2× from algorithmic improvement, highlighting the multiplicative benefit of hardware–algorithm co-design. Together, these results establish ADMs as a scalable and general, hardware-aligned framework for low-latency and energy-efficient generative modeling on analog computing platforms.

Keywords: generative AI, analog computing, diffusion models, flow matching, optical computing, neuromorphic computing, fixed-point compute paradigm

Generative artificial intelligence (GenAI) has emerged as one of the most transformative technological developments of recent years, driving advances in language modeling, visual media generation, and scientific discovery. Within this landscape, diffusion models[21,22] have become the dominant paradigm for generating continuous data modalities, including images[23], audio[24], video[25,26], and scientific data[27,28]. Despite their empirical success, diffusion models typically require many sequential evaluations to iteratively transform noise into data, making inference compute-, energy- and latency-intensive.

Efforts to improve the efficiency of diffusion models have followed two largely independent directions: alternative hardware designs and algorithmic advances. In this context, particular emphasis is on the inference stage, which is estimated to account for 90% of the energy consumed in commercial deployments[29].

Firstly, as digital electronics approach physical and economic limits, analog and optical computing[4–20] have re-emerged as promising alternatives, offering orders-of-magnitude improvements in latency and energy efficiency for compute intense matrix multiplications central to AI applications. However, analog hardware is susceptible to noise and often comes with a limited set of operations different from what is required for today's models which are optimized for digital accelerators. Therefore translating

these gains to the application level remains an open challenge. Existing optical implementations of diffusion models use diffractive optical layers and hybrid analog–digital systems[11,12]. This limits them to either linear architectures, due to the absence of inter-layer nonlinearities[11], or to designs dominated by digital computations[12]. Previous analog electronic implementations using resistive and in-memory computing platforms have demonstrated accelerated diffusion inference[13,14]. However, these efforts rely on non-standard diffusion formulations and training workflows, preventing the use of conventional models and established pipelines, and thereby limiting scalability, generality and adoption. To our knowledge, no existing diffusion framework simultaneously supports scalable multi-layer architectures, fully analog inference, and compatibility with well-established training procedures, three properties that are key to a general, efficient, and scalable diffusion system.

Secondly, substantial algorithmic efforts have focused on reducing the number of sampling steps required for high-quality generation, either by distilling or rectifying the model[30,31] or improving sampling schemes[32–34]. In the latter case, most existing approaches rely on low-order explicit discretizations of the underlying ordinary differential equation (ODE), such as Euler, Heun, or midpoint schemes[33]. While these methods are computationally efficient, requiring only one or two function evaluations per step, they can become unstable when the vector field varies rapidly with its inputs. For diffusion this is the case particularly near the data manifold and close to pure noise[35]. In such regimes, explicit solvers often require increasingly small step sizes to maintain stability and may produce overly smooth solutions when uncertainty is high[33]. Implicit solvers, which rely on implicit discretizations of the underlying ODE, offer a promising alternative in these settings, but have seen limited adoption in diffusion models since their iterative nature causes overheads when deployed on conventional digital hardware. This motivates implementing implicit diffusion sampling on analog hardware, where physical equilibration efficiently realizes the iterative

computations. The impact of analog noise, a central challenge of such non-traditional accelerators, is mitigated both by the attractor dynamics of the fixed-point iterations and by the inherent denoising nature of the diffusion process. Importantly, to ensure that the resulting algorithms can be broadly used across models, datasets, and hardware platforms, rather than being confined to specialized formulations, such an approach must remain fully compatible with standard diffusion training pipelines. We achieve this through careful co-design of the algorithms with the inherent properties of the hardware, without sacrificing generality.

Here, we introduce Analog Diffusion Models (ADMs), a diffusion framework that realizes diffusion steps through the physical equilibrium dynamics of analog hardware. During inference, ADMs employ an implicit ODE solver, evaluating each diffusion step via a fixed-point search. Crucially, training remains identical to that of conventional diffusion models and does not require fixed-point search. This sets ADMs apart from native fixed-point approaches such as deep equilibrium models[36] and their diffusion variants[37–39], which require fixed-point search during training, increasing computational cost and complexity and reducing generality.

The fixed-point search required by ADMs for inference can be implemented on a range of analog platforms, either through energy-minimizing physical systems[40,41] or via architectures that implement iterative computations, involving matrix multiplications and nonlinear operations in a closed feedback loop[20,42–44]. These iterative architectures can be realized on analog hardware such as optical, resistive, memristive, or neuromorphic systems, provided they support these core operations[5,13,14,18,19,42,43]. We adopt the iterative architecture[20] and perform analog diffusion inference on an Analog Optical Computer (AOC). AOC performs fixed-point search entirely in the analog domain by combining three-dimensional optics and analog electronics in a loop until convergence (Fig. 1a). By avoiding repeated digital–analog conversions and

merging compute and memory, such non-traditional hardware can achieve substantial efficiency gains of up to three orders of magnitude over digital accelerators[5,18–20].

With ADMs, training is performed on digital graphics processing units (GPU), which conventional diffusion training is optimized for. To account for non-idealities of the analog system, we train our models using a digital twin (DT) of the hardware, which applies hardware specific alterations and corruptions to the parameters and outputs of a model. The model itself is kept in standard form and the digital twin can be seen merely as a differentiable translation layer to the simulated hardware output for given standard neural network. Unlike some prior approaches that incorporate hardware-in-the-loop training or iterative digital-twin refinement within the learning loop[11,45–47], we employ a single, task-agnostic DT whose non-model parameters are kept fixed across all experiments. This preserves standard diffusion training while accounting for hardware non-idealities in the forward pass, and eliminates the need for hardware evaluations during training. This also allows for a wide range of hardware implementations, including those that are unsuitable for backpropagation based training[45].

We benchmark ADMs across a range of experimental settings. First, we demonstrate fully analog diffusion sampling on an AOC hardware with 2,304 optical weights using diverse two-dimensional (2D) distributions (Fig. 1b, Fig. 2a). To our knowledge, this constitutes the first demonstration of multi-layer diffusion models operating entirely on scalable analog hardware while maintaining a standard training pipeline. Second, we extend to larger image generation via a latent diffusion framework[48]. Executing the iterative diffusion process entirely on AOC and performing only the final decoding digitally, we generate MNIST, FashionMNIST (FMNIST), and ExtendedMNIST (EMNIST) samples (Fig. 1c). Third, we study the algorithmic benefit and limitations of ADMs at scale using a large-scale DT and common backbone architectures, including convolutional U-Nets[49] and diffusion transformers (DiT)[50], on

high-resolution image datasets, such as Smithsonian Butterflies, AFHQ, and Oxford Flowers-102. Across all architectures, datasets and model sizes, ranging from 32 million to 13 billion parameters, ADMs and its variants consistently reduce the number of diffusion steps required to reach a given sample quality. In particular, they achieve the same Fréchet Inception Distance (FID) as conventional diffusion models with up to 16× fewer steps.

Importantly, ADMs translate the entire low-level, per-operation efficiency gains of next-generation analog hardware into application-level improvements. By trading off diffusion steps and fixed-point iterations, ADMs combine the demonstrated algorithmic improvements with the projected analog hardware energy gains to indicate a potential >100× improvement in energy efficiency, or equivalently latency, relative to standard diffusion sampling running on digital hardware at the same generation quality (Fig. 1d and Discussion). These results establish ADMs as a general, efficient, scalable and low-latency route towards fully-analog generative modeling.

Analog realization of fixed-point diffusion steps

Diffusion models iteratively transform Gaussian noise into samples from a target distribution by integrating either a stochastic or an ordinary differential equation[35,51]. Here we focus on ODE-based diffusion variants, specifically the recently proposed flow-matching formulation[31,51,52]. Our method extends directly to other deterministic sampling methods like Denoising Diffusion Implicit Models[32].

While the iterative nature of ODE solvers is conceptually compatible with analog hardware, direct implementation faces two mismatches. First, diffusion models integrate their underlying ODE over a finite time horizon, typically from $t = 0$ to $t = 1$, whereas analog systems evolve according to physical relaxation dynamics toward a stationary state, effectively solving the ODE as $t \to \infty$. Second, early termination at a precise time is generally difficult on analog systems: unlike digital implementations

with well-defined discrete iterations, many analog platforms[13,14,20] evolve through continuous-time dynamics without an intrinsic notion of iteration count. To address these issues, we implement the diffusion process using an implicit ODE solver, where each step corresponds to a fixed-point search. The choice of ODE solver affects the number of steps required to reach a given sample quality[33,53]. While conventional diffusion solvers employ explicit discretizations of the underlying ODE (Eq. (3)), ADMs rely on an implicit discretization scheme (Eq. (4)), requiring the solution of a nonlinear system at each step. This formulation naturally aligns with equilibrating analog hardware, defining each diffusion step as the stable fixed-point of a learned nonlinear transformation conditioned on the previous state and diffusion time.

This formulation avoids the stability limitations of explicit discretizations in rapidly varying vector field regimes near the data manifold and pure noise[35]. It further provides robustness to analog noise, arising from two complementary mechanisms. First, the attractor dynamics of the fixed-point iteration make each diffusion step inherently robust to noise (Supplementary Fig. 4a). Second, the diffusion process itself further suppresses noise, since each denoising step is designed to operate on noisy inputs, with additional analog noise effectively absorbed into the diffusion noise (Supplementary Fig. 4b-c).

Model parameters are learned using standard diffusion objectives on digital GPUs, without fixed-point search during training. Because analog non-idealities affect inference-time behavior, in analog models training and inference are inherently coupled: to reflect the non-idealities correctly, inference needs to be based on exactly the same operations as training, which in this case would require prohibitively costly fixed-point training. We avoid this by capturing these non-idealities in the DT and exploiting a duality between explicit and implicit Euler discretization schemes. This ensures the competing goal of consistency between training and inference while preserving standard training procedures, yet enables implicit ODE discretizations in ADMs that

are naturally suited to equilibrating analog hardware with inherent noise. Details on methodology and implementation are provided in the Methods.

While ADMs can be implemented across diverse analog hardware systems, we demonstrate their operation on AOC that integrates 3D optical and analog electronic technologies in a recursive loop. We use micro light-emitting diode (microLED) arrays, reflective spatial light modulators (SLMs) with pixels encoding model weights, and photodetector (PD) arrays, leveraging fan-out and fan-in of light in the third dimension to enable inherently parallel, scalable matrix multiplications. Compared to the system presented in [20], our hardware is scaled up by a factor of about 10×, to 2,304 programmable optical weights, and has been re-designed (e.g. implementing two independent optical paths instead of a single pass to represent positive and negative weights simultaneously) to reduce noise, mitigate non-idealities, and minimize discrepancies between experimental results and DT (refer to Extended Data Fig. 1 and Methods for additional hardware details). Fig. 1a describes the modular design for implementing multi-layer diffusion models on non-traditional hardware: each layer of the neural network is represented by a module, implementing optical matrix-multiplication and analog non-linearity. Multiple modules are connected sequentially through a crossbar analog switch, and the final module is connected back to the first to form a closed loop for the implicit scheme. The same principle can be implemented using a single module with one of the blockmatrix structures exemplarily shown in the inset. The latter approach has been used in our experiments. Representative 2D distributions generated on hardware are shown on the right. Fig. 1b provides a detailed visualization of the diffusion process using hardware results. Each diffusion step consists of a fixed-point search (columns). Once converged, the result of the diffusion step is used to condition the next fixed-point search (rows). On this system, fixed-point inference for 2D distributions takes 10–15 μs per diffusion step, governed by the current system bandwidth of 5 MHz (-3dB; Supplementary Fig. 1b). Such per-step latencies are expected to be

on the order of nanoseconds in future hardware generations with target modulation speeds in the tens of GHz[4,54].

Fig. 2a presents samples from a diverse set of 2D distributions generated on the same hardware, showing that analog fixed-point search reliably implements diffusion steps despite hardware noise and device nonidealities. Across the 2D target distributions, the hardware results closely match the DT results, with a per-step normalized root mean square error (NRMSE) around 10^{-2} (Supplementary Fig. 2c). For quantitative benchmarking, we compare distributional discrepancies of the generated distributions to the target distribution with those of a Gaussian distribution used as a metric reference (Fig. 2b and c; *Supplementary Section D* for metric definitions). In all cases, our results significantly improve over the Gaussian reference. To reduce remaining stochastic errors, we average results either across multiple experimental runs (Extended Data Fig. 7a) or over convergence windows (Extended Data Fig. 7b), which yields stable outputs. Due to the noise robustness, five repetitions corresponding to convergence window of 0.2 μs is sufficient.

For completeness, in addition to ADMs, our methodology enables the execution of conventional (explicit) diffusion ODE steps on analog hardware (*Supplementary Section C.2*). Explicit schemes are not governed by attractor dynamics unlike implicit schemes, and are therefore more sensitive to analog noise. In the following we will focus on the implicit version for its better hardware alignment and superior efficiency.

Latent ADMs on AOC

Following the state-of-the-art in diffusion modeling, we employ a latent diffusion framework[48], performing the diffusion process in the latent space of a variational autoencoder (VAE). At training time, an ADM learns the latent space distribution of the VAE-encoded data. At inference time, the latent ADM is evaluated on the hardware and the final result mapped to the data space using the VAE decoder (Fig. 3a).

The diffusion process runs entirely in the analog domain, while only the final decoding step remains digital. In latent diffusion at scale, energy consumption is dominated by the iterative diffusion process in latent space, whereas VAE decoding is a single one-time operation, making the diffusion process the key bottleneck and particularly well suited for analog acceleration. The all-analog nature of our diffusion process is in contrast to prior optical diffusion work[12], where a large digital decoder network is employed in every diffusion step, leaving only a small fraction of the diffusion process to be implemented in the optical domain.

Using the latent ADM approach we generate samples from the canonical MNIST, FMNIST and EMNIST datasets. Fig. 3b compares samples generated with ADM on the AOC hardware with those obtained using ADM on the DT and a standalone VAE (VAE-alone). Results show that ADMs (hardware) substantially outperform the VAE-alone baselines qualitatively (Fig. 3b) and quantitatively, yielding lower FID in the data space (Fig. 3c) and lower distributional discrepancies in the latent space (Fig. 3d). Extended Data Fig. 8 provides additional representative results for EMNIST, showing the evolution of the ADM (hardware) samples across diffusion steps and the robustness to analog noise across trials and time intervals. Supplementary Fig. 5 further shows close correspondence between the latent distributions generated by the hardware, the DT and the ground truth on MNIST. These benchmarks demonstrate that latent ADMs provide a hardware-compatible pathway for high-dimensional generative modeling with our prototype hardware, enabling high-fidelity image synthesis while confining all iterative diffusion computations to the analog domain.

Analog Diffusion at Scale

For benchmarking beyond the current hardware scale, we evaluate the implicit sampling scheme of ADMs on multiple high-dimensional image datasets comprising Smithsonian Butterflies, AFHQ, and Oxford Flowers-102, and across a range of model

architectures. These include convolutional U-Nets, DiT, and a scaled-up DT of our analog hardware without non-idealities. In addition to the implicit Euler scheme discussed so far (labeled as Implicit), we investigate the Crank-Nicolson solver (Implicit-CN), an implicit variant that combines the implicit update with an additional explicit model evaluation, enabling direct compatibility with our hardware (see Methods). We compare these approaches with the explicit Euler scheme (Explicit) employed in state-of-the-art models[26,55]. We consider an image resolution of 256×256 and employ a standard latent diffusion approach using the frozen Stable Diffusion VAE[26] and models with sizes in the range between 32 million and 13 billion parameters. All our models are trained using the standard flow matching objective without loss weighting or rectification steps. For each experiment the same model is evaluated with different schedulers. Architectural details and training hyperparameters are provided in Supplementary Section A.3.

Across datasets and backbone architectures, implicit sampling consistently outperforms explicit sampling in the low-step regime, while converging to the same asymptotic sample quality as the number of diffusion steps increases (Fig. 4a). For example, on Oxford Flowers-102 using a U-Net backbone, the implicit sampler, and similarly Implicit-CN, achieves an FID of 26.5 using only 8 diffusion steps (with 4 fixed point iterations per step), outperforming the explicit variant with 128 steps (FID 30.6), corresponding to a 16× reduction in steps and 4× reduction in total neural network evaluations (algorithmic improvement), see middle graph in Fig. 4a. On Smithsonian Butterflies, using the DT of our hardware, both implicit variants achieve a smaller but consistent reduction in steps (4×) compared to explicit, and up to 2× algorithmic improvement, see left most graph in Fig. 4a. On AFHQ with DiT, the implicit sampler shows no advantage. This is explained by the stability of the fixed-point search. Stable convergence imposes a lower bound on the number of diffusion steps, $T_{\min}$, which scales with the Lipschitz constant of the learned vector field v, $T_{\min} \sim \mathrm{Lip}(v)$, see Theorem

1 in Method. In the case of AFHQ with DiT, this constraint limits the effectiveness of basic implicit methods. Here, the Crank–Nicolson variant, Implicit-CN, provides a more favorable trade-off. As a second-order method with $\mathcal{O}(\Delta t^2)$ global error, compared to $\mathcal{O}(\Delta t)$ for explicit and implicit Euler, it improves accuracy at larger step sizes, while its fixed-point formulation admits a more relaxed stability condition, with $T_{\min,\mathrm{CN}} \sim 2\times\mathrm{Lip}(v)$. On AFHQ, where the Implicit scheme is unstable at low diffusion step counts, Implicit-CN restores stable fixed-point convergence and recovers the efficiency gains of implicit sampling. This yields an up to 8× reduction in diffusion steps, comparing 8 steps of Implicit-CN (FID 19.2) with 64 steps of Explicit (FID 18.7) and 1-2× reduction in total neural network evaluations, see most right graph of Fig. 4a.

Fig. 4b shows qualitative samples comparing implicit variants (Implicit for Smithsonian Butterflies and Oxford Flowers-102, and Implicit-CN for AFHQ) with the explicit Euler scheme at low diffusion step counts across datasets and architectures. For the same number of steps, implicit sampling yields sharper images with clearer structure and greater sample diversity, whereas the explicit baseline exhibits noticeable blur and loss of detail. These qualitative differences are consistent with the corresponding improvements observed in FID.

Considering both, the number of diffusion steps and fixed-point iterations, combined with projected energy efficiencies of analog hardware, we investigate the potential energy efficiency of implicit solvers. Fig. 4c shows FID versus projected energy for different implicit solver configurations on the butterfly dataset compared to conventional explicit Euler sampling on digital hardware, each for 8, 16, 32 and 64 diffusion steps respectively. A detailed examination of these projections is deferred to the Discussion.

Extended Data Fig. 2 shows that implicit samplers approach convergence with as little as two fixed-point iterations; more iterations can improve the results especially for small numbers of diffusion steps but show diminishing improvement for 8 and

more steps. In particular for eight or more diffusion steps implicit sampling with two iterations yields similar or better results than a higher order Heun sampler, which is a commonly used best practice[33] and uses the same number of function evaluations. Additional qualitative results are presented in Extended Data Fig. 3, Extended Data Fig. 5, Extended Data Fig. 4.

Finally, Fig. 4d and Extended Data Fig. 6 qualitatively show that the implicit paradigm can be beneficial even for highly optimized state-of-the art models. We use FLUX.1[55], a 13 billion parameter text-to-image diffusion model. As a rectified flow model optimized for few-step explicit Euler sampling, the model produces high quality results already at 5 explicit steps. To explore a possible advantage of implicit methods we thus reduce the number of steps to 2. The standard implicit scheme does not converge at such low step counts, as per previous discussion. However, Implicit-CN achieves robust fixed-point convergence: with 2 diffusion steps and 4 fixed-point iterations, it significantly improves sample quality, producing sharper fine-grained details. These initial findings underline the generality of the approach, extending beyond controlled experiments to industry-grade diffusion models.

Taken together, our results demonstrate that implicit diffusion samplers constitute a general and practically effective alternative to explicit schemes, with clear benefits that persist from medium-scale image models to highly optimized frontier systems.

Discussion

In this work, we introduce ADMs, a diffusion framework that co-designs implicit solvers with non-traditional hardware to address three key challenges: generality, efficiency, and scalability, for diffusion atop non-traditional hardware.

By confining fixed-point search to inference while retaining standard diffusion training, ADMs overcome the training challenges typically associated with fixed-point models such as deep equilibrium models, while harnessing the energy efficiency

and noise robustness of analog fixed-point dynamics, which complement the noise-suppressing properties of diffusion denoising. This ensures broad applicability and generality of ADMs.

We demonstrate the feasibility of ADMs on a 2,304-optical-weight AOC, generating diverse 2D distributions, and extending the approach via latent diffusion to enable implicit sampling on AOC for benchmarking across several larger datasets. In digital simulations, we benchmark ADMs across a range of high-dimensional image generation tasks, model backbones and sizes. Across all settings, implicit variants consistently reduce diffusion steps by up to 16× and network evaluations by 1-2× while maintaining asymptotic sampling quality. While implicit sampling allows using fewer diffusion steps, its reliance on fixed-point convergence imposes a minimum number of diffusion steps dependent on the backbone model's Lipschitz constant. This can lead to failure modes for the implicit Euler method, as observed on AFHQ using DiT and FLUX. The Crank-Nicolson method (Implicit-CN), a hardware compatible implicit variant that combines higher-order accuracy with relaxed convergence constraints, overcomes this limitation, consistently producing high-quality samples with stable convergence, and achieves an algorithmic reduction of required function evaluations of approximately 2×. These results position ADMs as a promising pathway toward more compute-efficient diffusion sampling, even atop digital hardware, and motivate further investigation.

The main advantage of ADMs is that they retain operation-level efficiencies of analog hardware at the application level and further compound these with algorithmic improvements. Estimated analog efficiencies reach up to femtojoule-per-operation or peta operations per second per watt[5,18–20], suggesting potential energy savings of 100× to 1000× relative to current digital accelerators. Assuming a conservative 100× per-operation energy improvement and the 1-2× algorithmic advantage achieved with our proposed hardware model architecture, ADMs yield projected energy savings,

or equivalent latency reductions, of more than 100× versus explicit digital sampling (Fig. 4d). These gains arise from the co-design of our method with hardware properties, effectively translating low-level analog efficiency into application-level improvements.

Realizing these gains for industrial and scientific applications requires analog hardware that matches the scale and precision requirements of modern generative models. Advances in low-bit and quantized inference on digital hardware[56] are expected to carry over to analog implementations, further increasing their efficiency. Scaling is particularly critical for high-dimensional image generation tasks, which demand millions or billions of weights. Significant progress has recently been demonstrated in this direction[4,7], but planar optical approaches[4,7,10,18] remain limited by chip reticle size, as routing and computation share the same plane. In contrast, the AOC hardware we used for the ADM implementation leverages the third dimension to separate computation from data routing, enabling improved scalability. Each SLM pixel can serve as an individual weight modulator, with a footprint that is several orders of magnitude smaller than in planar alternatives. Off-the-shelf SLMs support up to four million μm-size pixel arrays allowing to represent matrices of sizes up to 1000×1000 per module, while emerging metasurface arrays promise 9.5 million pixel with microsecond-scale refresh rates[54], opening new scaling regimes for optical accelerators. Although current 3D prototypes rely on chip-sized technologies and bulky systems[12,19,20], future platforms will require full integration of the optical stack within centimeter-scale volumes. The modular design in Fig. 1a enables layers of up to four million parameters with current SLMs, comparable to the feedforward layers in modern models[55]. Larger layers can be achieved by distributing the optical matrix-vector multiplications over replicated modules. Finally, realizing full hardware efficiency requires analog memory, e.g., a sample-and-hold circuit, to avoid digital conversions between diffusion steps.

Overall, the results presented here point toward a new trajectory for GenAI. By grounding diffusion inference in the relaxation of a physical system rather than digital

iteration, ADMs offer a general and scalable route to high-throughput, energy-efficient and low-latency generative modeling. As analog accelerators mature and expand in scale, this hardware–algorithm alignment will be a promising path to reduce the computational and environmental burden of GenAI workloads.

Methods

Flow matching and definitions

Sampling from a flow-matching based diffusion model corresponds to integrating the ODE

$$\frac{d}{dt}x(t) = f_\theta(x(t), t), \qquad x(0) = x_0, \tag{1}$$

from $t = 0$ to $t = 1$, where $x_0 \sim N(0, I)$ is a sample from a multi-variate Gaussian and in our case $f_\theta : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ denotes a time-conditional, K-layer feedforward neural network of the form

$$f_\theta(x, t) = \big(L_K \circ \cdots \circ L_1\big)(x, t), \qquad L_i(x, t) = W_i\sigma(x) + b_i + te_i. \tag{2}$$

Each layer is a map $L_i : \mathbb{R}^{d_{i-1}} \times \mathbb{R}_+ \to \mathbb{R}^d$, with weight $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$, bias $b_i \in \mathbb{R}^{d_i}$, and time embedding vector $e_i \in \mathbb{R}^{d_i}$, $i = 1, \ldots, K$, and $d_0 = d_K = d$. The trainable model parameters are $\theta := \{W_1, b_1, e_1, \ldots, W_K, b_K, e_K\}$. A common ODE solver is the explicit Euler discretization, which for a fixed number of steps $T = \frac{1}{\Delta t}$ iterates the update

$$x_{t+\Delta t} = x_t + \Delta t\, f_\theta(x_t, t). \tag{3}$$

In contrast the implicit Euler scheme uses the implicit update rule

$$x_{t+\Delta t} = x_t + \Delta t\, f_\theta(x_{t+\Delta t}, t + \Delta t), \tag{4}$$

which, as $x_{t+\Delta t}$ appears on both sides, requires solving a system of non-linear equations at each step. Importantly, both updates use the same operator $\text{Euler}_{x_t,t,\Delta t,\theta}(z) := x_t + \Delta t\ f(z, t; \theta)$ merely evaluated at different $t \in [0, 1]$ and $z \in \mathbb{R}^d$.

ADMs: Methodology

The idea behind ADMs is to construct a hardware compatible operator, which can be evaluated both explicitly during training (Eq. (3)) and implicitly during inference (Eq. (4)). In the following we will derive this construction.

ADMs are implemented by embedding the network (2) into a higher-dimensional analog state space $\mathbb{R}^n$ and realizing its computation through repeated application of a fixed update rule. We denote by

$$\mathrm{ADM}_{x,t;\theta}(z) = W\sigma(z) + b + te + \mathrm{pad}(x) \tag{5}$$

the ADM operator, a re-parameterization of the update implemented by AOC, where $z \in \mathbb{R}^n$ is the internal AOC state, $b, e \in \mathbb{R}^n$ are concatenations of biases b_i and time embeddings e_i respectively, and $\mathrm{pad}(x) := (x, 0, \ldots, 0)^\top \in \mathbb{R}^n$ embeds the diffusion state $x \in \mathbb{R}^d$ into the analog state space $\mathbb{R}^n$. AOC iterates this update until convergence to a fixed-point $z^* \in \mathbb{R}^n$, with $\mathrm{ADM}_{x,t;\theta}(z^*) = z^*$.

The purpose of the digital twin is to incorporate the non-idealities expected at inference time during training. However, this makes it crucial that training and inference rely on the same update operator. The below construction of the ADM operator ensures this property.

Blockmatrix layout

Multi-layer ADMs can be implemented either combining multiple AOC systems in a modular way (Fig. 1a), where all modules participate in a shared recurrent loop, or by using a single AOC system with the block structure shown in the bottom of Fig. 1a (implicit). Both implementations are mathematically equivalent and we will focus on the latter in the following.

The block-structured matrix W and bias b of ADMs are constructed such that successive applications of the ADM operator correspond to cycling the signal through the layers of the network. Specifically, with each application of the ADM operator the signal gets propagated down the state and the output of the last layer gets fed back to the top. For $\Delta t = \frac{1}{T}$ we define $\theta(\Delta t) = \{W_{\Delta t}, b_{\Delta t}, e_{\Delta t}\}$ with

$$W_{\Delta t} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \Delta t\, W_K \\ W_1 & 0 & \cdots & 0 & 0 \\ 0 & W_2 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & W_{K-1} & 0 \end{bmatrix}, \quad b_{\Delta t} = \begin{bmatrix} \Delta t\, b_K \\ b_1 \\ b_2 \\ \vdots \\ b_{K-1} \end{bmatrix}, \quad e_{\Delta t} = \begin{bmatrix} \Delta t\, e_K \\ e_1 \\ e_2 \\ \vdots \\ e_{K-1} \end{bmatrix}, \tag{6}$$

as well as $\mathrm{pad}(x) = (x, 0, \ldots, 0)^\top \in \mathbb{R}^n$.

Training

Setting $\Delta t = 1$ at training time, after K applications of the ADM operator, the first d components (defined as $[\cdot]_{:d}$) of the AOC state recover the network output. In contrast to evaluation in the hardware which always iterates the operator until convergence to a fixed-point, during training on GPU using the DT we can evaluate the system after a finite number of ADM iterations. This means at training time we can evaluate the model explicitly via

$$f_\theta(x, t) = \Big[\big(\mathrm{ADM}_{x=0,t;\theta(1)}\big)^K(x) \Big]_{:d}, \tag{7}$$

where ADM^K denotes K applications of the ADM operator, and thus use standard (explicit) training procedures.

Using this formulation, model parameters θ are learned using standard conditional flow matching with optimal transport path [51]. Given samples $x_0 \sim p_0$ (Gaussian base distribution) and $x_1 \sim p_1$ (data distribution), as well as timestep $t \sim \mathrm{Uniform}([0, 1])$

and the linear interpolation $x_t = tx_1 + (1-t)x_0$, the training objective is

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{(x_0,x_1),t} \left\| f_\theta(x_t, t) - (x_1 - x_0) \right\|_2^2, \tag{8}$$

with f_θ as defined in (7).

Inference

To correctly account for the non-idealities of the hardware, at inference time we can only apply the same operator learned during training. By construction of the ADM operator, K applications of $\text{ADM}_{x_t,t;\theta(\Delta t)}$ on x_t yield the explicit Euler update (3), i.e.,

$$\left[\text{ADM}^K_{x_t,t;\theta(\Delta t)}(x_t)\right]_{:d} = \text{Euler}_{x_t,t,\Delta t;\theta}(x_t) = x_t + \Delta t \; f_\theta(x_t, t). \tag{9}$$

The underlying working principle of ADMs is that iterating this update rule (with $t \to t + \Delta t$) beyond K iterations is still meaningful as the fixed-point of the Euler operator provides the implicit Euler update (4) and as the Euler operator itself can be represented by K ADM iterations, consequently the fixed-point of the ADM operator can be identified with the implicit Euler update. This is formalized by the following theorem.

Theorem 1 (ADM Inference) *Let σ be $\text{Lip}(\sigma)$-Lipschitz and let $\|\cdot\|_\infty$ denote the induced matrix norm. If $\Delta t < \frac{1}{\text{Lip}(\sigma)^K \|W\|_\infty^K}$, then for any (x,t) the operator $\text{ADM}_{x,t;\theta(\Delta t)}$ has a unique fixed point $z^* \in \mathbb{R}^n$ and, for any initialization z_0, we have $\lim_{k\to\infty} \text{ADM}^k_{x,t;\theta(\Delta t)}(z_0) = z^*$. Moreover, the readout components $[z^*]_{:d}$ of the fixed-point solve the implicit Euler equation (4).*

Details and proofs can be found in *Supplementary Section C*.

Finally, combining these aspects yields our training and inference process described in Algorithm 1.

The same duality between explicit and implicit Euler scheme can be directly applied to make the Crank-Nicolson solver, whose update rule is given by $x_{t+\Delta t} = x_t + \Delta t/2\,(f_\theta(x_t, t) + f_\theta(x_{t+\Delta t}, t + \Delta t))$, compatible with analog hardware. To this end, first evaluate $\tilde{x}_t = x_t + \Delta t/2\ f_\theta(x_t, t)$ either in digital or using our analog version of explicit diffusion sampling (see *Supplementary Section C.2*). Using this pre-computed $\tilde{x}_t$, the result of the Crank-Nicolson scheme is given by the fixed-point of the operator $\mathrm{ADM}_{\tilde{x}_t, t+\Delta t; \theta(\Delta t/2)}$, which is immediately compatible with AOC. Note that convergence of the Crank-Nicolson solver is already guaranteed for the less restrictive condition $\Delta t < \frac{2}{\mathrm{Lip}(\sigma)^K \|W\|_\infty^K}$, which translates to a $2\times$ smaller minimum number of steps and allows more robust few-step sampling as demonstrated by the results on AFHQ in Fig. 4.

Relaxed fixed-point iterations to address hardware constraints

Optical vector–matrix multiplication (OVMM) requires fanning-out and fanning-in optical signals to and from all the modulators, i.e., the pixels of the SLM in AOC, representing the weights. As matrices become increasingly sparse, the total reflected or transmitted optical power decreases, making the system progressively more lossy. In the extreme case of an identity matrix, all optical power is attenuated except at a single modulator per row and column, rendering such operations impractical in the optical domain. To solve this general optical hardware limitation, we propose a novel methodology that is implemented in AOC as follows. We implement an analog skip connection that bypasses the OVMM for a fraction α of the signal, while the remaining $(1-\alpha)$ fraction is processed through the lossy optical path. During training, such a skip connection would break the duality between explicit and implicit Euler schemes that underlies ADMs. However, when applied as a post-processing step to a trained model, this mechanism corresponds to a relaxed fixed-point iteration and can be exploited without modifying the training procedure. Specifically, we define an

Algorithm 1 Training and inference of ADMs

Require: Network parameters $\theta_{\Delta t}$, step size Δt, total diffusion steps T
Require: (Digital twin of) ADM operator $\mathrm{ADM}_{x,t;\theta}$

Training (explicit evaluation using digital twin)
for each minibatch **do**
  Sample $x_0 \sim p_0$ (Gaussian), $x_1 \sim p_1$ (data)
  Sample $t \sim \mathcal{U}[0,1]$
  Interpolate $x_t \leftarrow t x_1 + (1-t)x_0$
  Evaluate network explicitly: ▷ K-layer model

$$f_\theta(x_t, t) \leftarrow \left[(\mathrm{ADM}_{x=0,t;\theta_{\Delta t=1}})^K (x_t) \right]_{:d}$$

  ▷ $[\cdot]_{-d:}$ for explicit
  Compute loss ▷ Standard CFM loss

$$\mathcal{L} \leftarrow \| f_\theta(x_t, t) - (x_1 - x_0) \|_2^2$$

  Update parameters θ via backpropagation and gradient descent
end for

Inference (implicit evaluation using hardware):
Initialize $x_0 \sim p_0$
for $n = 1, \ldots, T$ **do**
  Set $t_n \leftarrow n/T$ ▷ $t_n \leftarrow (n-1)/T$ for explicit
  Initialize AOC state $z^{(0)}$ arbitrarily
  repeat
    $z^{(k+1)} \leftarrow \mathrm{ADM}_{x_n, t_n; \theta_{\Delta t}}(z^{(k)})$ ▷ Fixed-point search
  until convergence to fixed point z^* ▷ Theorem 1 guarantees convergence
  Extract denoised update: ▷ Take first d entries ($[\cdot]_{-d:}$ for explicit)

$$x_{n+1} \leftarrow [z^*]_{:d}$$

end for
return x_T

α-relaxed fixed-point iteration as

$$\mathrm{ADM}^{(\alpha)}_{x,t;\theta}(z) := \alpha \odot z + (1-\alpha) \odot \mathrm{ADM}_{x,t;\theta}(z), \qquad 0 \le \alpha < 1, \tag{10}$$

which introduces an analog skip connection while preserving the structure of the ADM operator. Crucially, after convergence the relaxed operator provides the same result as the original ADM operator, since the mappings $x \mapsto f(x)$ and $x \mapsto \alpha x + (1-\alpha) f(x)$ admit identical fixed-points. Under the same step-size conditions as in the non-relaxed case, the α-relaxed operator converges to a unique fixed point corresponding to the

diffusion update (Theorem 6). The skip connection can therefore be introduced post-training through a rescaling of the learned model parameters and only increases the number of fixed-point iterations required for convergence.

By selecting α appropriately, either globally or in a channel-wise (per analog state variable) manner, we can trade off optical power loss against the number of fixed-point iterations. This enables the representation of large-weight matrices within the dynamic range of the hardware while maintaining stable convergence and an improved effective signal-to-noise ratio. Further details are provided in *Supplementary Section C.3*.

Analog hardware platform

Experiments were conducted on an analog optical computing platform comprising 2,304 programmable weights implemented through 3D optical matrix multipliers integrated with channel-wise analog electronic nonlinearities in a recursive architecture (see Extended Data Fig. 1 for further details). During each fixed-point iteration, the system state vector z_k switches back and forth between optical and analog electronic forms. Matrix multiplications occur in the optical domain, where z_k is encoded as light intensities from microLED arrays (peak wavelengths 503 nm at 20 mA), and the weight matrix W is encoded onto reflective SLM pixels. Light emitted from each microLED fans out across a row of the SLM for element-wise multiplication, with resulting signals summed column-wise by a PD array as electrical voltages[57]. Building upon the optical matrix multiplication scheme described in[20], we implement separate optical paths for positive and negative weights, each using an independent SLM, to reduce system crosstalk. This scheme leverages 3D optics, with efficient fan-in and fan-out enabled by spherical and cylindrical lenses, enabling inherently parallel and scalable large-matrix multiplications. Analog voltages are processed through channel-wise nonlinear activation functions (hyperbolic tangent), implemented with bipolar differential pairs. The custom electronic board design is detailed in[20] and Extended Data Figure 3 therein. Each analog iteration involve three steps: nonlinear activation,

matrix multiplication, and summation with fixed bias, time-embedding, and conditioning voltages. The same weights are reused at every iteration, following an in-memory computing style. The system runs until the signals stabilize, reaching a fixed point for each respective diffusion step. Finally, the voltages are recorded using a 48-port digitizer (Spectrum M2p.5913).

Sampling and experiment workflow

Waveforms were measured with a sampling rate of 4.6875 MHz. Each experiment captured 90 samples, corresponding to a total acquisition window of 19.2 µs. The fixed-point search remained entirely in the analog domain during this interval. Samples were collected between 68% and 93% of the solve time, which translates to a time window from 13.1 µs to 19.7 µs. This range ensures that equilibrium dynamics are captured after initial transients but before full convergence. Once sampled, readout voltages were linearly converted to solution variables. Voltages were averaged over 100 experiment for figures in the main text; averaging over 5 experiments shows comparative generation quality (see Extended Data Fig. 7 for 2D distributions and Extended Data Fig. 8c-d for latent tasks).

DT calibration procedure

The DT is calibrated in two stages: an open-loop calibration that identifies component-level hardware parameters, and a closed-loop calibration that refines system-level dynamics. In the open-loop stage, we measure the behavior of each physical block—electrical offsets and noise characteristics, the analog nonlinearity, microLED input efficiency, SLM response, photodiode gain and crosstalk, and polarization-beam-splitter leakage—using dedicated experiments designed to isolate each contribution. Together, these component-level effects comprise the principal hardware nonidealities discussed in the ADM methodology; characterizing them enables defining the initial parametric form of the DT. To capture remaining linear distortions, we perform an

open-loop OVMM sweep over 100 random weight matrices, 6 input vectors, and 7 values of the gradient gain β (which scales the optical signal applied to the $W\sigma(z)$ term in (5)), and fit output-correction vectors. Across 4200 open-loop experiments, the mean NRMSE was 0.015 (Supplementary Fig. 2a). This modular calibration workflow is general in structure and is intended to be adaptable, in principle, to other analog computing platforms comprising linear-nonlinear building blocks.

In the closed-loop stage, we validate and adjust the DT under recurrent operation. Using 1000 experiments that span random weight matrices, channel-wise nonlinear coefficients, and realistic β configurations, we refine a small set of global scaling parameters: the gains on the gradient, input, and the α skip connection term described in (10), along with residual offsets. This yields a digital model that accurately reproduces the hardware's equilibrium trajectories across configurations used in the diffusion experiments. Mean NRMSE across 1000 closed-loop experiments was 0.009 (Supplementary Fig. 2b).

Acknowledgments

We acknowledge colleagues at Microsoft Research, Cambridge, UK for fruitful discussions.

Author contributions

H.K., F.F., and J.C. contributed equally to this work. J.C. and F.P. are corresponding authors. J.G., F.P. and H.B. conceived and managed the project. H.K. and F.F. developed the framework and performed the hardware simulations for the 2D experiments. H.K. derived the algorithm and theoretical results and performed the large scale simulations. F.F. performed the hardware simulations for the latent experiments. J.C., J.H.C., D.J.K., D.C., G.B. and F.P. proposed, designed and implemented the optical

and electrical parts. J.C. performed the experiments, analysed their data and implemented the digital twin with support from J.H.C., D.J.K., D.C., G.B. and F.P. D.J.K., J.H.C., F.P., K.P.K. and D.C. developed the scaling roadmap. J.C., D.J.K., J.H.C., developed the control software. B.R., K.P.K., F.P., C.G., P.J., B.C., M.H., S.R., V.R., J.K. and H.B. provided technical input to the simulations and experiments. H.K., J.C., F.P., F.F., J.H.C. and K.P.K wrote the paper with input from all authors.

Competing interests

The authors of the paper have filed several patents relating to the subject matter contained in this paper in the name of Microsoft Co.

Code availability

The code used in this study is currently being prepared for public release and will be made available upon publication. During the review process, the code is available from the corresponding author upon reasonable request.

Data availability

The data that support the findings of this study are currently being prepared for public release and will be made available upon publication. During the review process, the data are available from the corresponding author upon reasonable request.

Additional information

Correspondence and requests for materials should be addressed to Jiaqi Chu or Francesca Parmigiani.

Figures

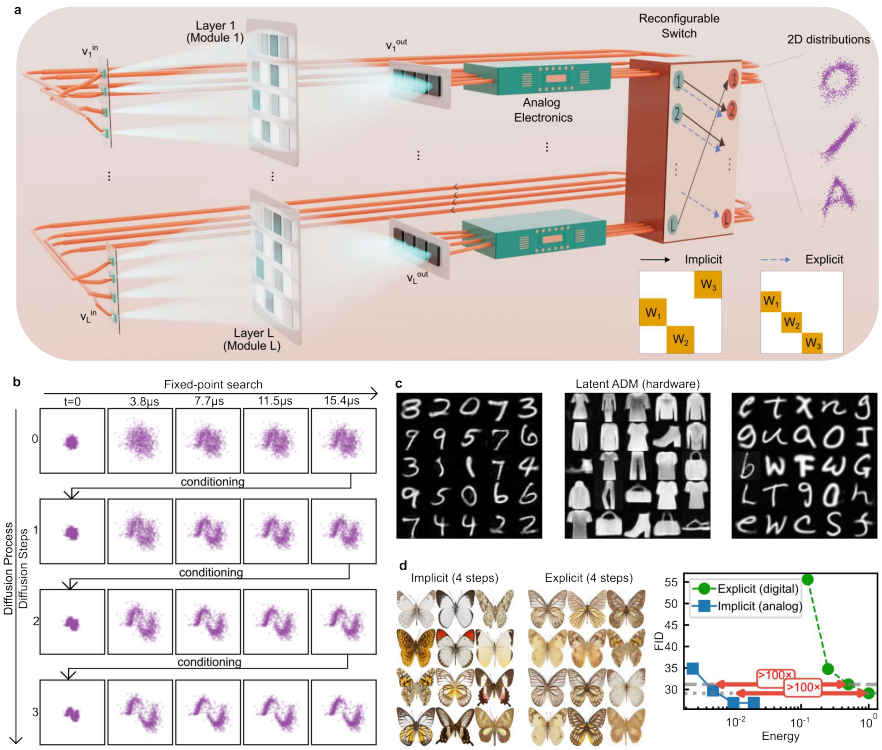


Fig. 1: Analog Diffusion Models (ADMs). **a,** ADMs implemented on modular analog optical computer (AOC) units. Multiple units are connected to implement the multi-layer structure of the diffusion model. A programmable switch selects between implicit and explicit evaluation modes. Mathematically, the multi-unit architecture is equivalent to a single system with a block-matrix layout, as illustrated in the inset. Each unit comprises optical vector–matrix multipliers implemented using a three-dimensional optical system with spatial light modulators, with nonlinearities provided by analog electronics. Conditioning information is injected into the loop as fixed voltages, and the recursive system converges to a fixed-point. Representative two-dimensional distributions generated by the hardware are shown on the right. **b,** The ADM sampling process proceeds along two axes: each diffusion step (rows) is implemented as a fixed-point search (columns) conditioned on the previous result (results on hardware). **c,** Selected latent ADM samples generated on the hardware for MNIST, FMNIST and EMNIST. **d,** Simulation of Implicit (analog) versus Explicit (digital) on the Smithsonian Butterflies dataset using the digital twin (DT) of the hardware. Qualitative images at 4 diffusion steps and quantitative FID versus projected energy for the two solvers, illustrating the excess of 100× application-level energy efficiency gain.

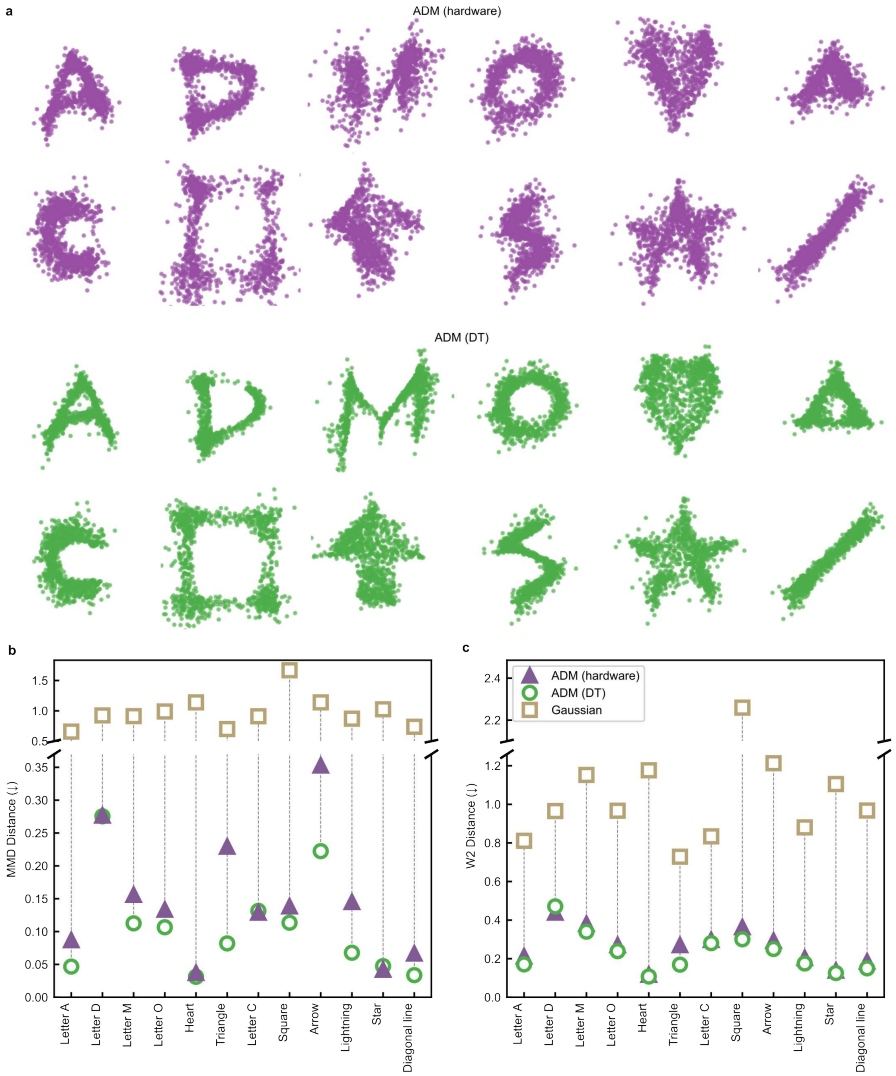


Fig. 2: ADM inference for 2D distributions. **a,** Examples of 2D distributions generated by ADMs on the hardware (top two rows) and its DT (bottom two rows). From left to right, we show: A, D, M , O, heart, triangle (top row), C, square, arrow, lightning, star and diagonal line (bottom row). **b,** Maximum Mean Discrepancy (MMD) between the ground truth and distributions generated by the hardware, the DT and a Gaussian reference distribution. The reference distribution is obtained by fitting a 2D multivariate Gaussian to the dataset points. All 2D point clouds are normalized to the square [-1, 1]^2 using the dataset distribution for each shape as reference. **c,** Wasserstein-2 (W2) distance between the groundtruth and generated distributions for multiple 2D datasets.

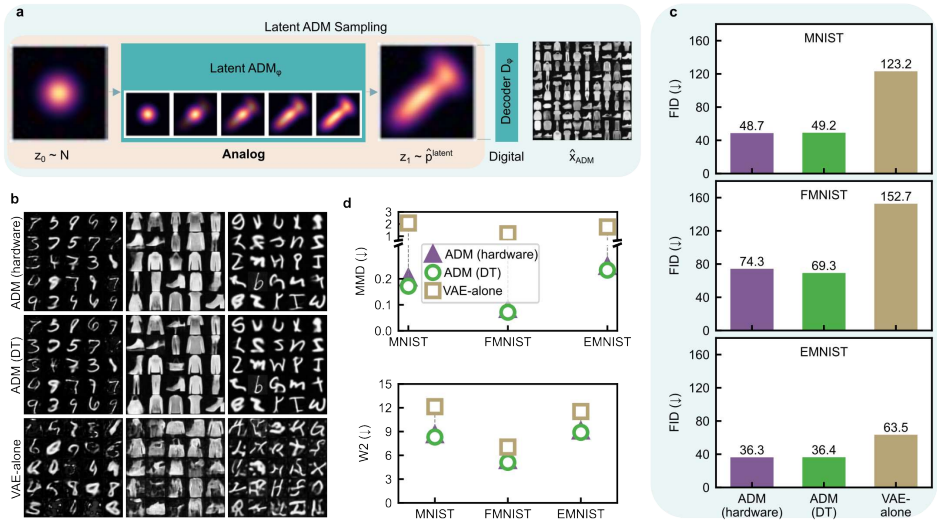


Fig. 3: Latent ADMs: operating principle, characterization and results. **a,** Schematic of the latent ADM framework. All diffusion steps are fully computed in analog hardware, and only the final decoding step is performed digitally. **b,** Samples generated by the latent ADM for MNIST (left), FMNIST (middle) and EMNIST (right), comparing ADMs on hardware (top), ADMs on the DT (middle) and the VAE-alone samples (bottom). **c,** Fréchet Inception Distance (FID) for MNIST (top), FMINST (middle) and EMNIST (bottom), computed using 1,000 generated samples, comparing ADMs on hardware, ADMs on the DT, and the VAE-alone, highlighting the contribution of the ADM to the image generation. **d,** Latent-space distribution similarity between generated and target data (MMD and W2). Hardware results (triangles) closely match DT (circles), while VAE-alone (squares) serves as a baseline.

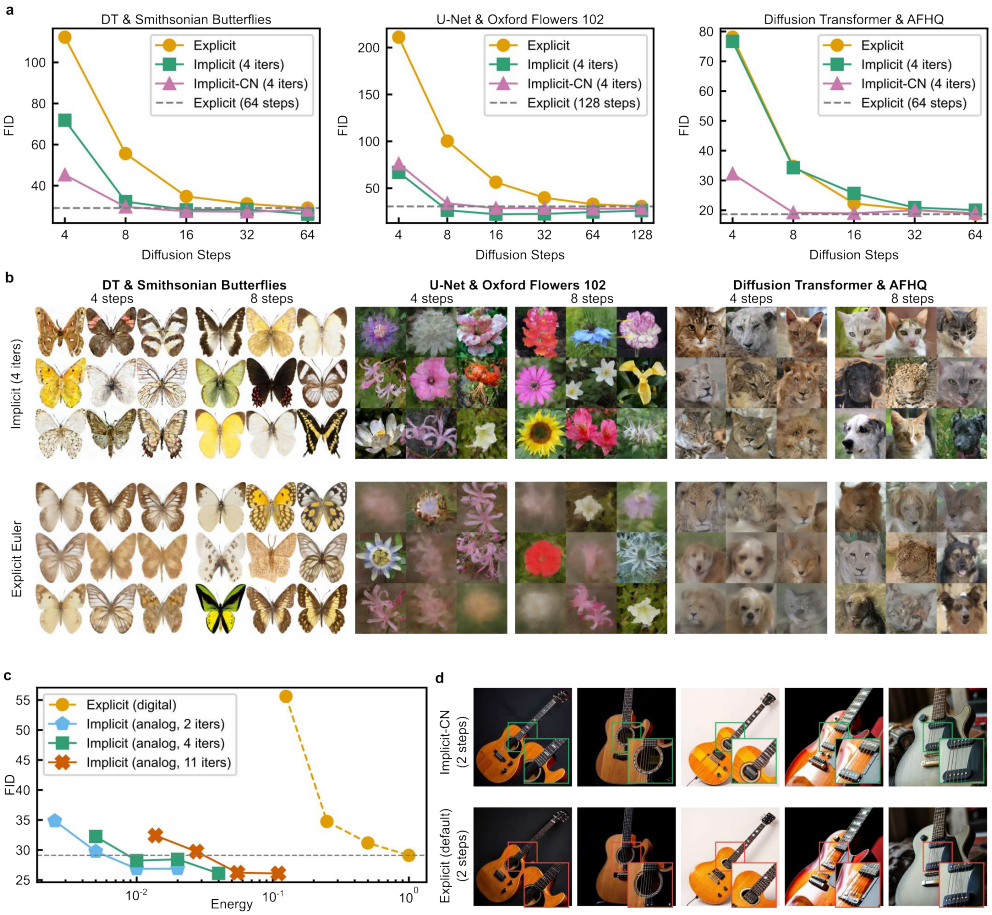
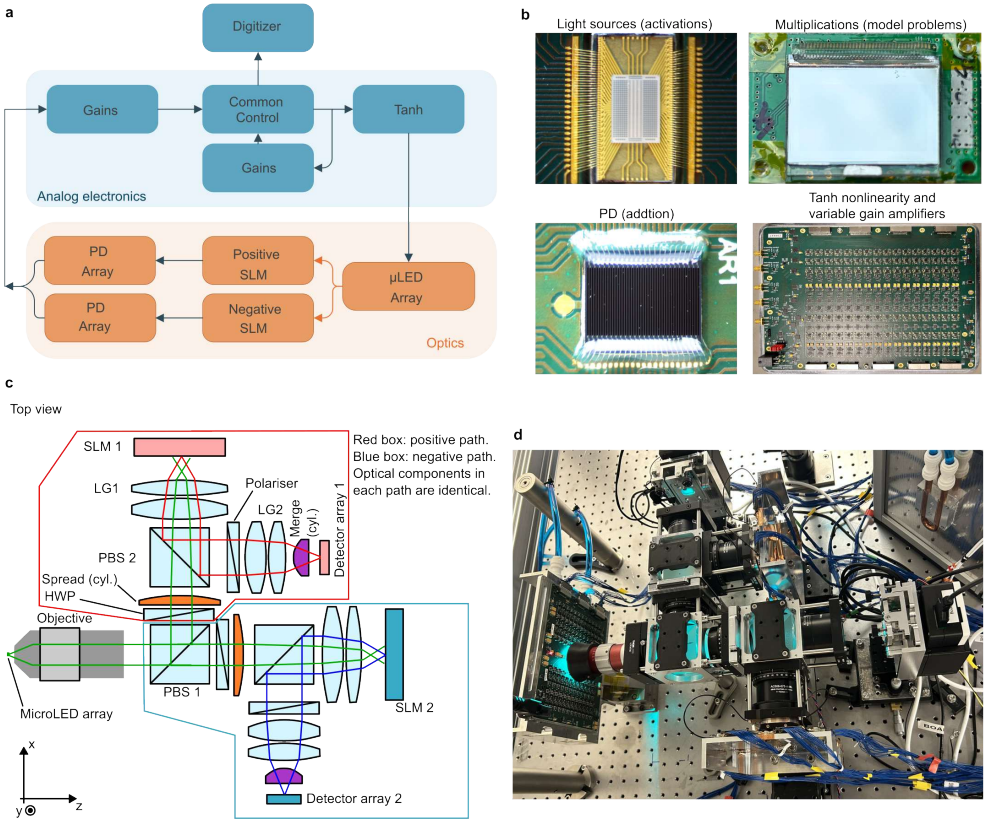
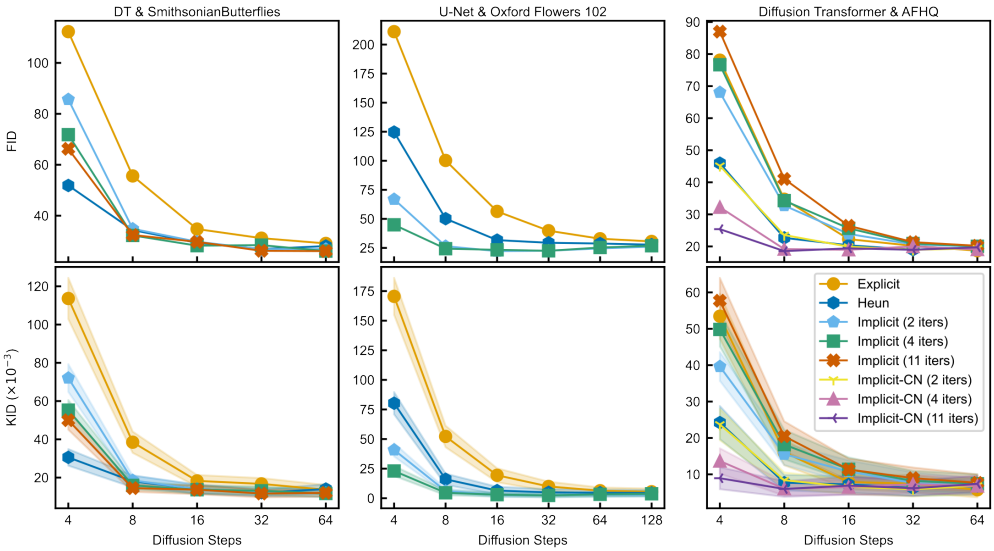


Fig. 4: Benchmarking ADMs at scale. **a,** FID versus number of diffusion steps for variants of implicit (this work) and explicit (standard) sampling from the same diffusion models for multiple datasets and model architectures. Implicit models use 4 fixed-point iterations per diffusion step. **b,** Qualitative comparison of sampling schemes at low step numbers. Butterflies (DT) and Flowers (U-Net) are generated with the standard Implicit method and AFHQ (DiT) shows the Implicit-CN variant. **c,** Trade-off study between sample quality and energy consumption: each curve represents the FID at 8, 16, 32 and 64 diffusion steps respectively, versus the normalized energy for different numbers of fixed-point iterations per diffusion step. **d,** Qualitative demonstration of Implicit-CN sampling with the state-of-the art text-to-image diffusion model FLUX.1 [55] using the prompt "a guitar".

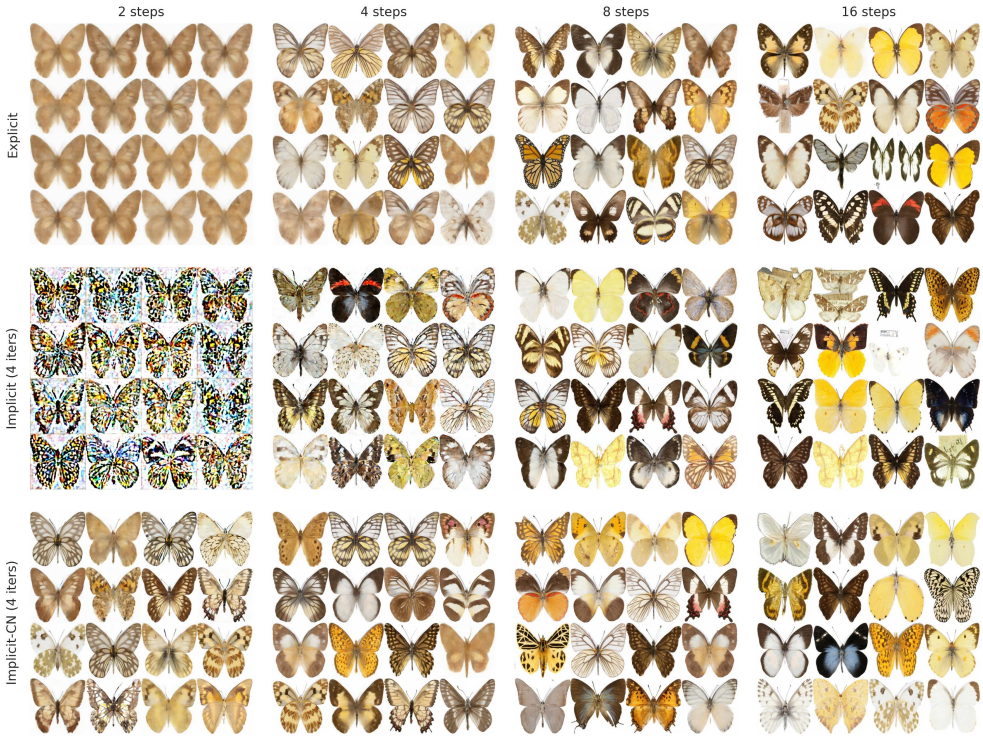
Extended Data Figures



Extended Data Fig. 1: Hardware architecture of the analog computing platform and corresponding key components. **a,** Schematic diagram of the analog electronic (in blue) and the optical (in orange) sub-systems, implementing the fixed-point update rule (Eq.1). Arrows are shown to indicate the signals' flow. **b,** Key hardware components photos. The 4x48 microLED array (top left) is the light source and represents neural network activations. The spatial light modulator (SLM) (top right) stores neural network weights, and multiplies them with the incoming light. The 48 photodetector array (bottom left) adds and transfers optical signals into the analog electronic domain. The nonlinearity, programmable variable gains, and other computations are applied in analog electronics. **c,** Schematic of the optical subsystem, illustrating dual optical paths for positive and negative weights. The sources, collected by the objectives, are imaged onto the SLM using the combined 4F system of the objective and lens groups, respectively. **d,** Overhead view of the optical analog hardware.



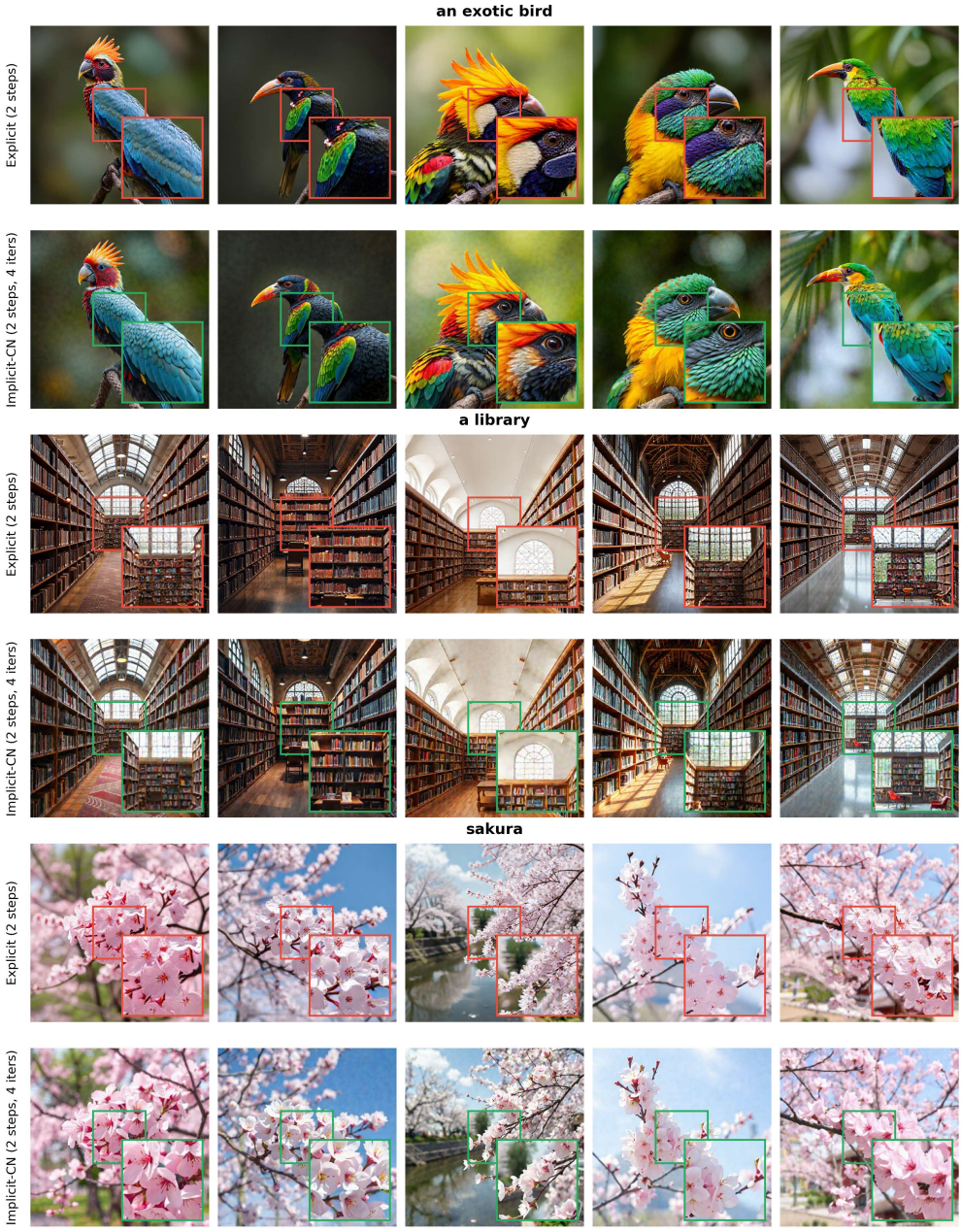
Extended Data Fig. 2: FID and Kernel Inception Distance (KID) for additional sampler configurations. For 8 or more diffusion steps 2 fixed-point iterations can be sufficient to reach convergence of the implicit solvers. Implicit sampling can yield the same (on Butterflies with $\geq$ 8 steps) or better performance (on Flowers) with the same number of neural network evaluations compared to a higher order Heun solver, which is a common best practice [33]. Implicit-CN with 2 fixed-point iterations is mathematically identical to the Heun scheme and additional iterations can provide further improvements as shown for AFHQ. In all experiments we use a warm start of the fixed-point search, by initializing at the explicit Euler result, which requires one network evaluation and hence is counted as one fixed-point iteration.



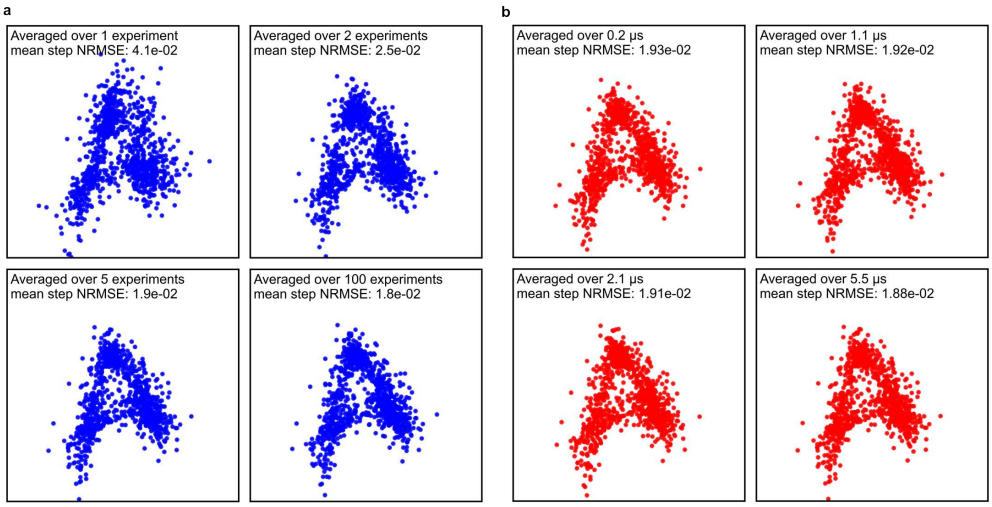
Extended Data Fig. 3: Additional samples for the three schemes (Explicit, Implicit and Implicit-CN) from the Digital Twin trained on the Smithsonian Butterflies dataset.

Extended Data Fig. 4: Additional samples for the three schemes (Explicit, Implicit and Implicit-CN) from the U-Net trained on Oxford-Flowers102.

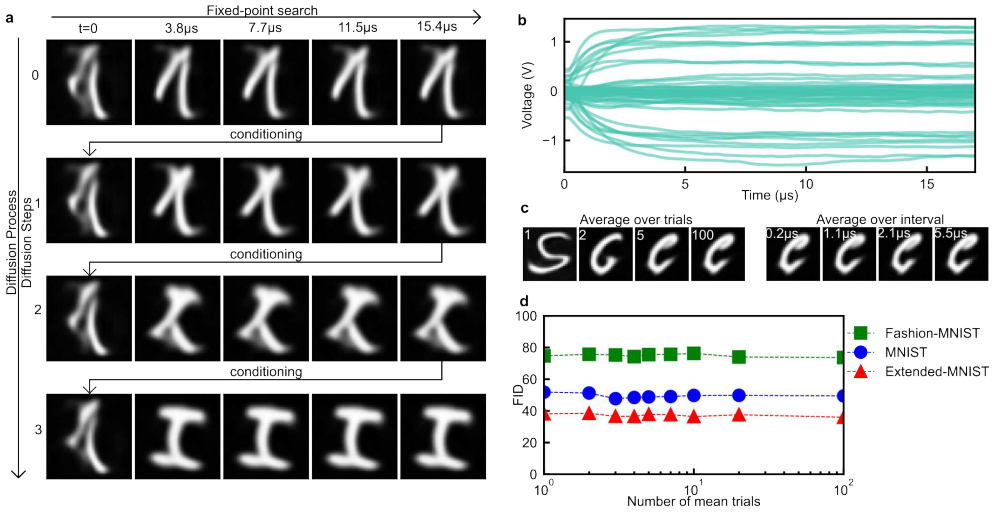
Extended Data Fig. 5: Additional samples for the three schemes (Explicit, Implicit and Implicit-CN) from the Diffusion Transformer (DiT) trained on AFHQ.



Extended Data Fig. 6: Additional samples for different prompts produced with the state-of-the-art text-to-image diffusion model FLUX.1 [55] using explicit Euler (default) and Implicit-CN sampling with 2 diffusion steps and 4 fixed-point iterations. The respective prompts are contained in the titles.



Extended Data Fig. 7: Additional noise details on the 2D experiments. **a,** Effect of averaging across repeated 2D experiments for the letter A, chosen as a representative example, for ADMs generated on AOC. Increasing the averaging beyond 5 yields negiligible additional reductions in NRMSE. **b,** Effect of averaging within a fixed-point time window for the letter A for ADMs generated on AOC. Increasing the averaging window (up to 5.5 µs) yields small but consistent reductions in NRMSE, consistent with SLM-dominated noise.



Extended Data Fig. 8: Inference dynamics for latent ADM hardware experiments on EMNIST, chosen as a representative example among the considered datasets. **a,** Evolution of latent ADM inference for EMNIST. Each row shows the progression of the fixed-point search within a diffusion step, followed by successive diffusion steps (rows). **b,** Example traces of the 48 analog voltages over time for a representative fixed-point within a diffusion step. **c,** Effect of averaging across repeated experiments and within a fixed-point time window. **d,** FID as a function of the number of mean trials for MNIST, FMNIST and EMNIST, showing negligible improvement beyond five averages across datasets.

References

[1] Thompson, N. C., Greenewald, K., Lee, K., Manso, G. F. *et al.* The computational limits of deep learning. *arXiv preprint arXiv:2007.05558* **10**, 2 (2020).

[2] PILZ, K. F., MAHMOOD, Y. & HEIM, L. Ai's power requirements under exponential growth (2025).

[3] Horowitz, M. *et al.* Scaling, power, and the future of cmos 7–pp (2005).

[4] Ahmed, S. R. *et al.* Universal photonic artificial intelligence acceleration. *Nature* **640**, 368–374 (2025).

[5] Chen, Y. *et al.* All-analog photoelectronic chip for high-speed vision tasks. *Nature* **623**, 48–57 (2023).

[6] Dong, B. *et al.* Partial coherence enhances parallelized photonic computing. *Nature* **632**, 55–62 (2024).

[7] Hua, S. *et al.* An integrated large-scale photonic accelerator with ultralow latency. *Nature* **640**, 361–367 (2025).

[8] Leroux, N. *et al.* Analog in-memory computing attention mechanism for fast and energy-efficient large language models. *Nature Computational Science* 1–12 (2025).

[9] Zhang, Y. *et al.* Direct tensor processing with coherent light. *Nature Photonics* 1–7 (2025).

[10] Shen, Y. *et al.* Deep learning with coherent nanophotonic circuits. *Nature photonics* **11**, 441–446 (2017).

[11] Oguz, I. *et al.* Optical diffusion models for image generation. *Advances in Neural Information Processing Systems* **37**, 59150–59173 (2024).

[12] Chen, S., Li, Y., Wang, Y., Chen, H. & Ozcan, A. Optical generative models. *Nature* **644**, 903–911 (2025).

[13] Cheng, Y. *et al.* Voltage-controlled magnetoelectric devices for neuromorphic diffusion process. *Nature Communications* **16**, 5022 (2025).

[14] Yang, J. *et al.* Resistive memory-based neural differential equation solver for score-based diffusion model. *arXiv preprint arXiv:2404.05648* (2024).

[15] Shastri, B. J. *et al.* Photonics for artificial intelligence and neuromorphic computing. *Nature Photonics* **15**, 102–114 (2021).

[16] Feldmann, J., Youngblood, N., Wright, C. D., Bhaskaran, H. & Pernice, W. H. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature* **569**, 208–214 (2019).

[17] Wetzstein, G. *et al.* Inference in artificial intelligence with deep optics and photonics. *Nature* **588**, 39–47 (2020).

[18] Hamerly, R., Bernstein, L., Sludds, A., Soljačić, M. & Englund, D. Large-scale optical neural networks based on photoelectric multiplication. *Physical Review X* **9**, 021032 (2019).

[19] Wang, T. *et al.* An optical neural network using less than 1 photon per multiplication. *Nature Communications* **13**, 123 (2022).

[20] Kalinin, K. P. *et al.* Analog optical computer for ai inference and combinatorial optimization. *Nature* 1–8 (2025).

[21] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. & Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. *International conference on machine learning* 2256–2265 (2015).

[22] Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33**, 6840–6851 (2020).

[23] Baldridge, J. *et al.* Imagen 3. *arXiv preprint arXiv:2408.07009* (2024).

[24] Polyak, A. *et al.* Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720* (2024).

[25] DeepMind, G. Veo 3. https://deepmind.google/models/veo/ (2024).

[26] Blattmann, A. *et al.* Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* (2023).

[27] Lewis, S. *et al.* Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science* (2025).

[28] Zeni, C. *et al.* A generative model for inorganic materials design. *Nature* **639**, 624–632 (2025).

[29] Ivanov, A., Dryden, N., Ben-Nun, T., Li, S. & Hoefler, T. Data movement is all you need: A case study on optimizing transformers (2021). URL https://proceedings.mlsys.org/paper_files/paper/2021/file/bc86e95606a6392f51f95a8de106728d-Paper.pdf.

[30] Song, Y., Dhariwal, P., Chen, M. & Sutskever, I. Consistency models (2023).

[31] Liu, X., Gong, C. & Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003* (2022).

[32] Song, J., Meng, C. & Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).

[33] Karras, T., Aittala, M., Aila, T. & Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems* **35**, 26565–26577 (2022).

[34] Lu, C. *et al.* Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems* **35**, 5775–5787 (2022).

[35] Song, Y. *et al.* Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020).

[36] Bai, S., Kolter, J. Z. & Koltun, V. Deep equilibrium models. *Advances in neural information processing systems* **32** (2019).

[37] Geng, Z., Pokle, A. & Kolter, J. Z. One-step diffusion distillation via deep equilibrium models. *Advances in Neural Information Processing Systems* **36**, 41914–41931 (2023).

[38] Pokle, A., Geng, Z. & Kolter, J. Z. Deep equilibrium approaches to diffusion models. *Advances in Neural Information Processing Systems* **35**, 37975–37990 (2022).

[39] Bai, X. & Melas-Kyriazi, L. Fixed point diffusion models (2024).

[40] Syed, M., Kalinin, K. & Berloff, N. Beyond digital: Harnessing analog hardware for machine learning (2023).

[41] Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the national academy of*

sciences **81**, 3088–3092 (1984).

[42] Ambrogio, S. *et al.* An analog-ai chip for energy-efficient speech recognition and transcription (2023).

[43] Zoppo, G., Marrone, F. & Corinto, F. Equilibrium propagation for memristor-based recurrent neural networks (2020).

[44] Al-Kayed, N. *et al.* Programmable 200 gops hopfield-inspired photonic ising machine. *Nature* **648**, 576–584 (2025).

[45] Momeni, A. *et al.* Training of physical neural networks. *Nature* **645**, 53–61 (2025).

[46] Wright, L. G. *et al.* Deep physical neural networks trained with backpropagation. *Nature* **601**, 549–555 (2022).

[47] Rasch, M. J. *et al.* Hardware-aware training for large-scale and diverse deep learning inference workloads using in-memory computing-based accelerators. *Nature communications* **14**, 5282 (2023).

[48] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-resolution image synthesis with latent diffusion models 10684–10695 (2022).

[49] Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation (2015).

[50] Peebles, W. & Xie, S. Scalable diffusion models with transformers (2023).

[51] Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M. & Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747* (2022).

[52] Albergo, M. S. & Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571* (2022).

[53] Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T. & Mitliagkas, I. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080* (2021).

[54] Shebanow, M., Finkelstein, H. & Bowen, P. Revolutionizing the ai data center: Delivering 4,000 peta operations per second at 1% of the power using optical systolic arrays. White Paper, Neurophos Inc. (2025). URL https://11549dc2-0dc9-4363-9a47-d257d2a497cc.filesusr.com/ugd/79aa3c_85cd1a5477d44f4fa0e423c03ad0746c.pdf. Accessed: 2026-01-28.

[55] Labs, B. F. *et al.* Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742* (2025).

[56] Gadhikar, A., Grazzi, R., Hensman, J. *et al.* Optrot: Mitigating weight outliers via data-free rotations for post-training quantization (2024). URL https://openreview.net/forum?id=b5649238190ee2a0c8b8d7c25154a090163b6b4b.

[57] Farhat, N. H., Psaltis, D., Prata, A. & Paek, E. Optical implementation of the hopfield model. *Applied optics* **24**, 1469–1475 (1985).

[58] Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks (2010).

[59] Tolstikhin, I. O. *et al.* Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems* **34**, 24261–24272 (2021).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplementary.pdf