Arcee: Differentiable Recurrent State Chain for Generative Vision Modeling with Mamba SSMs

Jitesh Chavan[1] Rohit Lal[2,*] Anand Kamat[3,*] Mengjia Xu[1]
[1]New Jersey Institute of Technology [2]University of California, Riverside [3]McGill University
jsc78@njit.edu, rlal011@ucr.edu, anand.kamat@mail.mcgill.ca, mengjia.xu@njit.edu
*Equal contribution

Abstract

State-space models (SSMs), Mamba in particular, are increasingly adopted for long-context sequence modeling, providing linear-time aggregation via an input-dependent, causal selective-scan operation. Along this line, recent "Mamba-for-vision" variants largely explore multiple scan orders to relax strict causality for non-sequential signals (e.g., images). Rather than preserving cross-block memory, the conventional formulation of the selective-scan operation in Mamba reinitializes each block's state-space dynamics from zero, discarding the terminal state-space representation (SSR) from the previous block. Arcee, a cross-block recurrent state chain, reuses each block's terminal state-space representation as the initial condition for the next block ($h_0^{(l)} = \mathcal{T}^{(l)}(h_T^{(l-1)})$). Handoff across blocks is constructed as a differentiable boundary map whose Jacobian enables end-to-end gradient flow across terminal boundaries. Key to practicality, Arcee is compatible with all prior "vision-mamba" variants, parameter-free, and incurs constant, negligible cost. As a modeling perspective, we view terminal SSR as a mild directional prior induced by a causal pass over the input, rather than an estimator of the non-sequential signal itself. To quantify the impact, for unconditional generation on CelebA-HQ (256×256) with Flow Matching, Arcee reduces FID↓ from 82.81 to 15.33 (5.4× lower) on a single scan-order Zigzag Mamba baseline. Extensible CUDA kernels and training code are released to support reproducibility and further research at https://github.com/JiteshChavan/rc2

1. Introduction

Flow matching and diffusion models have revolutionized generative frameworks for images, videos, protein structures, and many other modalities (Lipman *et al*., 2022 [21]; Albergo *et al*. 2023 [1]; Liu *et al*. [24]; Bose *et al*., 2024 [6]; Song *et al*. [32] 2020; Karras *et al*. [19]). These models generate realistic images, videos or samples from corresponding data distribution by simulating an ordinary or stochastic differential equation (ODE/SDE) with a sample from a simple prior (usually gaussian noise) as the initial value condition for the differential equation, where the vector field (and score function in case of SDEs) that defines the differential equation is approximated by a neural network. Recently, transformer architectures have proliferated as a choice for the neural network, a consequence of their superior scalability [3, 30] and effectiveness in multi-modal training [4]. Despite their effectiveness for in-context learning tasks in non sequential modalities, transformers bear a significant computational cost that scales quadratically with input sequence length. While there have been efforts to alleviate the quadratic complexity of the attention mechanism by instrumenting methods such as FlashAttention, FlashAttention 2 [7, 8], it still remains the bottleneck for employing transformer-based models [35].

State-Space Models have emerged as competitive architectures for long context sequence modeling, offering linear time information aggregation across input signals via continuous time State-Space transitions [13, 14, 16]. Recent work improves SSM robustness and efficiency through better initializations [15], parametrizations [14], diagonalizations [16], and recurrence parallelizations [12]. *Mamba* [11] in particular extends prior works, making SSMs more expressive with input dependent state-space transitions through hardware-aware and work-efficient selective scan, yielding linear scaling in sequence length. While selective scan mechanism introduced in Mamba excels at efficient long sequence modeling, its causal aggregation of information creates friction when adapting Mamba to non sequential modalities, motivating architectures that preserve efficiency while relaxing the inherent strict causality. Prior vision-SSM work typically flattens 2D signals into a token sequence u and applies multiple scan orders within the same block, followed by simple feature fusion, adding parameters for each scan order [23, 25, 27, 38]. A complementary strategy amortizes layerwise heterogeneous scan orders across

depth with no per-block parameter increase, as in *Zigma* [17], where each causal scan manifold $u \mapsto y$, captures dependencies between tokens at varying degrees of spatial vicinity across layers.

Mamba was originally introduced for *autoregressive* sequence modeling; consequently, most vision variants inherit a design in which each block's state–space dynamics are initialized with a *zero* state instead of retrieving global causal summary encoded within terminal state-space representation h_T from previous block to avoid information leakage and preserve causality; This is restrictive for non-sequential signals (e.g., images), because h_T summarizes a full pass over the input in a given scan order and thus encodes potentially useful global context that is currently discarded.

In this work, we introduce a cross-block **R**ecurrent State **C**hain (**Arcee**) that generalizes the conventional causal selective-scan in Mamba by using the terminal state-space representation (SSR) from block $l-1$ as the initial value condition for SSM dynamics in block l. Any prior Mamba baselines can be adapted to propagate this compact global summary across depth, yielding a plug-and-play mechanism with zero additional parameters and constant, negligible overhead (independent of sequence length). Arcee is orthogonal to scan-order design; whereas standard selective-scan fixes each block's initial state to **0**, Arcee reuses the previous block's final SSR to provide a mild, architecture-agnostic inductive bias for generative visions tasks with Mamba SSMs. To summarize, we make following contributions:

1. We identify the legacy zero-init initial value condition for state-space dynamics in conventional selective scan operation in Mamba that discards the terminal state-space representation (SSR) between blocks (see 1, restrictive for non-sequential signals (images).
2. We hypothesize that the terminal SSR potentially encodes a useful global summary and serves as a mild inductive cue for downstream selective-scan dynamics, despite SSR being a severe compression of the non-sequential signal.
3. We introduce a zero parameter overhead, plug and play solution *Arcee* by generalizing the selective scan manifold in Mamba [11] from $u \mapsto y$ to $(u, h_0^{(l)}) \mapsto (y, h_T^{(l)})$ via a differentiable boundary map $h_0^{(l)} = \mathcal{T}^{(l)}(h_T^{(l-1)})$; its Jacobian $\mathcal{J}_{\mathcal{T}}^{(l)} = \partial h_0^{(l)} / \partial h_T^{(l-1)}$ is trained end to end so terminal SSRs $(h_T^{(k)} \forall k \in [0, depth))$ rendered by each block align across depth. In our default, $\mathcal{T}^{(l)}$ is identity.
4. Plugging Arcee into naive single-scan order ZigZag-Mamba baseline (Zigma_1) [17] for unconditional generation on CelebA-HQ (256x256) [18] with Flow Matching [21], we observe a large FID drop from **82.81** to **15.33** (-81.49%, **5.40x lower**) at negligible cost. As scan-order diversity increases (or in case of order-specific weights), the improvements attenuate; we analyze these trends in the experiments.
5. We release a general Arcee-selective-scan CUDA implementation with exposed boundary hooks (h_0, h_T) to enable research on cross-block state handoffs.

2. Background and Motivation

2.1. Generative Framework: Flow Matching

This work employs the Flow Matching framework (Lipman *et al*. 2022 [21]; Albergo *et al*. 2023 [1]; Liu *et al*. [24]) which learns a time–dependent vector field that transports a simple prior distribution P_{init} (e.g., $\mathcal{N}(0, I_d)$) to the data distribution p_{data} along the marginal probability path $P_t(x)$ by simulating an ODE.

We represent data points as vectors $z \in \mathbb{R}^d$ and the data distribution as p_{data}. The **Gaussian conditional probability path** $P_t(x_t \mid z)$ is constructed to interpolate between $\mathcal{N}(0, I_d)$ and $\delta_z(\cdot)$ on interval $t \in [0, 1]$:

$$\begin{aligned} x_t = \alpha_t z + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, I_d) \\ \iff \quad p_t(x_t \mid z) = \mathcal{N}\big(x_t; \alpha_t z, \sigma_t^2 I_d\big) \end{aligned} \tag{1}$$

where $\alpha_t, \sigma_t \geq 0$ are **interpolation schedulers** with $\alpha_0 = \sigma_1 = 0$ and $\alpha_1 = \sigma_0 = 1$ and α_t (resp. σ_t) strictly monotonically increasing (resp. decreasing). The conditional vector field for Gaussian conditional probability paths is given by (see [21, 22]):

$$u_t(x_t \mid z) = \frac{\dot{\sigma}_t}{\sigma_t} x_t + (\dot{\alpha}_t - \alpha_t \frac{\dot{\sigma}_t}{\sigma_t}) z \tag{2}$$

By construction, simulating an ODE with the conditional vector field $u_t(x \mid z)$ yields a trajectory that follows the Gaussian conditional probability path $p_t(x_t \mid z)$, thus the (Liouville [2]) continuity equation for the ODE described by $u_t(x_t \mid z)$ holds:

$$\begin{aligned} X_0 \sim \mathcal{N}\big(0, I_d\big), \quad \frac{\mathrm{d}}{\mathrm{d}t} X_t = u_t\big(X_t \mid z\big) \\ \implies X_t \sim p_t(\cdot \mid z) \quad \forall \quad t \in [0, 1] \\ \iff \partial_t p_t(x \mid z) = -\operatorname{div}_x \big(u_t(x_t \mid z) p_t(x_t \mid z)\big) \end{aligned} \tag{3}$$

Corresponding **marginal probability path**, $p_t(x_t) = \mathbb{E}_{z \sim p_{\text{data}}}[p_t(x_t \mid z)]$ is induced by making $z \sim p_{\text{data}}$ random. The marginal probability path interpolates Gaussian noise $P_0 \sim \mathcal{N}(0, I_d)$ and $p_1 = p_{\text{data}}$. Marginalizing (3) over $z \sim p_{\text{data}}$ yields the marginal continuity equation[22]:

$$\begin{aligned} X_0 \sim \mathcal{N}\big(0, I_d\big), \quad \frac{\mathrm{d}}{\mathrm{d}t} X_t = u_t(X_t) \\ \implies X_t \sim p_t \quad \forall \quad t \in [0, 1] \\ \iff \partial_t p_t(x) = -\operatorname{div}_x \big(u_t(x_t) p_t(x_t)\big) \end{aligned} \tag{4}$$

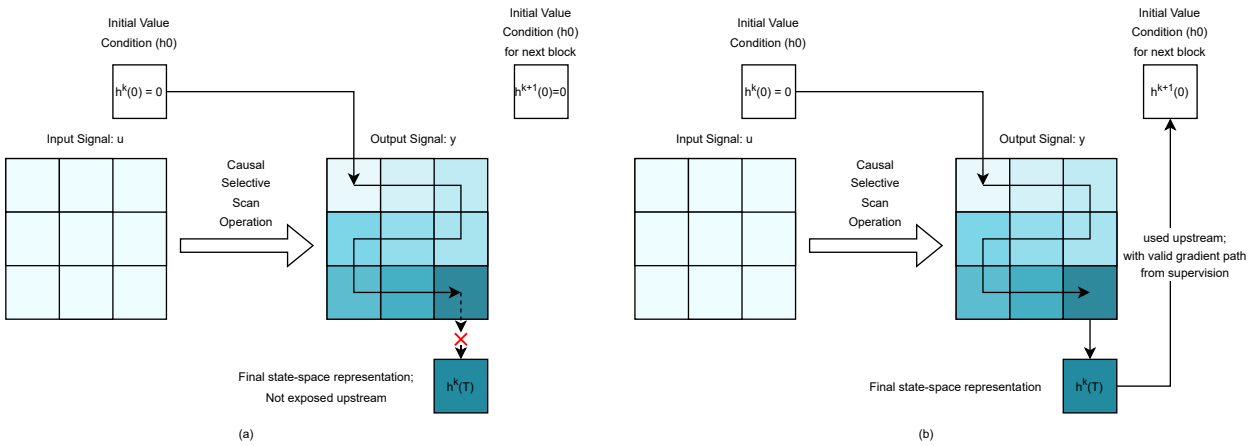

Figure 1. Conventional vs. Arcee selective scan. (a) In a vanilla Mamba block, the selective scan is strictly causal: the state is initialized with $h^{(k)}(0) = 0$, the terminal state $h^{(k)}(T)$ is discarded after producing y, and the next block again starts from zero. Darker cells indicate positions that have accumulated more context (later timesteps have seen a larger prefix of the sequence). (b) Arcee extends the scan to a two-port block: the terminal SSR $h^{(k)}(T)$ is reused as the initial state $h^{(k+1)}(0)$ of the next block via a differentiable boundary map, creating a recurrent state chain across depth with a valid gradient path and no change to the intra-block dynamics.

Flow Matching models learn to approximate the **marginal vector field** $u_t(x_t)$ using a deep neural network, Mamba backbone in particular for the scope of this paper:

$$u_t(x_t) = \int u_t(x_t \mid z)\, p_{1|t}(z \mid x_t)\, \mathrm{d}z,$$
$$p_{1|t}(z \mid x_t) = \frac{p_t(x_t \mid z) p_{\text{data}}(z)}{p_t(x_t)} \quad (5)$$

Conditional Flow Matching (CFM). Directly regressing the marginal field $u_t(x)$ via (5) is intractable because it requires the posterior $p_{1|t}(z \mid x_t) \propto p_t(x_t \mid z) p_{\text{data}}(z)$. Instead, CFM supervises the network with the *conditional* target $u_t(x_t \mid z)$, which admits a closed form under Gaussian probability paths (Eq. 2). We draw $t \sim \rho$, $z \sim p_{\text{data}}$, $\epsilon \sim \mathcal{N}(0, I)$ and set $x_t = \alpha_t z + \sigma_t \epsilon$, then minimize

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t\sim\rho, z\sim p_{\text{data}}, \epsilon}\left[\left\| u_\theta(x_t, t) - u_t(x_t \mid z) \right\|_2^2 \right]. \quad (6)$$

Training on the conditional target sidesteps the intractable posterior in the marginal formula. With an ℓ_2 loss, the best predictor at any x_t is the conditional mean of $u_t(x_t \mid z)$ over $z \sim p_{\text{data}}$, which equals the marginal vector field $u_t(x_t)$. i.e., CFM learns the *marginal* vector field without ever evaluating the intractable posterior $p_{1|t}(z \mid x_t)$. In practice we use $\rho(t) = \text{Uniform}[0, 1]$ and the GVP interpolant (α_t, σ_t) (Sec. 3).

Simulating an ODE with the marginal vector field from initial Gaussian noise leads to a trajectory whose marginals at p_t, such that $X_1 \sim p_{\text{data}}$, see Eq.(4), returns a sample from the desired distribution. This sampling is called **ODE sampling** with a flow matching model, analogous in spirit to ODE-based samplers developed for diffusion models [26].

2.2. Selective State Space Models: Mamba

From structured SSMs to Mamba. A continuous-time linear state-space model (SSM) evolves

$$\dot{h}(t) = A\, h(t) + B\, x(t), \qquad y(t) = C\, h(t) + D\, x(t), \quad (7)$$

where $h \in \mathbb{R}^{d_{\text{inner}} \times d_{\text{state}}}$, input $x \in \mathbb{R}^{d_{\text{inner}}}$, and output $y \in \mathbb{R}^{d_{\text{inner}}}$. Classical *structured* SSMs use fixed, specially parameterized (A, B, C, D) and model long-range dependencies within a sequential signal via a causal convolution kernel in discrete time.

Zero-order hold (ZOH) discretization. With a (possibly learned) positive step $\Delta > 0$ and zero-order hold assumption on the input over $[k\Delta, (k+1)\Delta)$, the exact discretization is

$$h_{k+1} = \bar{A}\, h_k + \bar{B}\, x_k, \qquad y_k = C\, h_k + D\, x_k, \quad (8)$$

where

$$\bar{A} = e^{\Delta A}, \qquad \bar{B} = \left(\int_0^\Delta e^{\tau A}\, d\tau \right) B. \quad (9)$$

Selective (input-dependent) SSM. Mamba [11] relaxes the time-invariance constraint in SSMs making the parameters B, C and step size Δ functions of input $x(t)$ at each

time step t, thereby making the state-space dynamics content aware

$$B_k, C_k, \Delta_k \;=\; \Phi(x_k\,\theta), \tag{10}$$

yielding a time-varying discrete recurrence that facilitates expressive causal aggregation of information across input signal

$$h_{k+1} = \bar{A}_k\, h_k + \bar{B}_k\, x_k, \qquad y_k = C_k\, h_k + D_k\, x_k, \tag{11}$$

with $\bar{A}_k = e^{\Delta_k A_k}$ and $\bar{B}_k = (\int_0^{\Delta_k} e^{\tau A_k}\, d\tau) B_k$.

2.2.1. Selective scan

Let $u \in \mathbb{R}^{T\times d_{\text{inner}}}$ be a discrete time input signal and assuming the selective scan at step t uses time variant SSM parameters $(\bar{A}_t = e^{\Delta_t A}, \bar{B}_t, C_t)$ (Eq. 10). The selective scan evaluates the causal recurrence

$$\begin{aligned} h_{t+1} &= \bar{A}_t h_t + \bar{B}_t\, u(t), \\ y(t) &= C_t h_t + D_t\, u(t), \\ h_0 &= 0\,. \end{aligned} \tag{12}$$

Unrolled recurrence. Expanding (12) yields

$$\begin{aligned} h_t &= \Big(\prod_{j=0}^{t-1} \bar{A}_j\Big) h_0 \;+\; \sum_{i=0}^{t-1} \Big(\prod_{j=i+1}^{t-1} \bar{A}_j\Big) \bar{B}_i u(i) \\ h_0 = 0 \implies h_t &= \sum_{i=0}^{t-1} \Big(\prod_{j=i+1}^{t-1} \bar{A}_j\Big) \bar{B}_i u(i), \end{aligned} \tag{13}$$

and therefore

$$y(t) = C_t h_t + D u(t) = D u(t) + C_t \sum_{i=0}^{t-1} \Big(\prod_{j=i+1}^{t-1} \bar{A}_j\Big) \bar{B}_i u(i). \tag{14}$$

where empty product is I (identity). Equation (14) implies that the discrete time output signal $y(t)$ depends only on $\{x(0), \ldots, x(t)\}$.

Hardware-aware evaluation. Rather than forming long products in (13), Mamba uses a fused prefix (monoid) scan on pairs $(\bar{A}_k, \bar{B}_k x_k)$ composed as $(\bar{A}', \bar{B}') \circ (\bar{A}, \bar{B}) = (\bar{A}'\bar{A},\ \bar{A}'\bar{B} + \bar{B}')$, yielding linear-time aggregation and high GPU efficiency.

2.3. Motivation and Method

It has been established that State Space Models are effective signal approximators [37]. In particular, due to their time variant selective scan operation, Mamba SSMs have shown promising results for efficient long sequence modeling tasks such as, tokenization free byte-based language modeling [36], modeling audio waveforms and DNA sequences [11].

2.3.1. Selective Scan Operation: Causality

Mamba aggregates information across input signal in an input dependent causal manner via the efficient selective scan operation, thus excelling at efficient long sequence modeling. Assume $u \in \mathbb{R}^{T\times d_{\text{inner}}}$ is the discrete time input signal and $y \in \mathbb{R}^{T\times d_{\text{inner}}}$ is the discrete time output of selective scan operation. Since from equation (14) it is evident that $y(t)$ depends only on $\{x(0), \ldots, x(t)\}$, we can formalize the conventional selective scan manifold (Fig. 1(a)), where state-space dynamics are always initialized with *0* initial value condition and terminal state-space representation h_T [1] is discarded after driving $y(T-1)$, as follows:

Conventional selective-scan manifold: Under (12)–(14) with the fixed initial condition $h(0) = \mathbf{0}$, a Mamba block implements a causal map

$$\mathcal{M}:\ u \mapsto y, \qquad y(t) = \mathcal{F}\big(u(0{:}t)\big), \quad t = 0, \ldots, T{-}1, \quad h_0 = 0 \tag{15}$$

and discards the terminal state-space representation h_T (otherwise denoted as $h(T-1)$) after producing $y(T-1)$ (Fig. 1(a)). Further, from (14) the Jacobian $J = \big[\partial y(i)/\,\partial u(j)\big]$

$$\frac{\partial y(j)}{\partial u(i)} = \begin{cases} D, & j = i, \\ C_j \Big(\prod_{k=i+1}^{j-1} \bar{A}_k\Big) \bar{B}_i, & i < j, \\ \mathbf{0}, & i > j, \end{cases}$$

is strictly lower-triangular, no input $u(i)$ can contribute to output $y(j)$ with $i > j$.

This causal aggregation of information causes friction when adapting Mamba to non-sequential signals (such as images), when flattened in particular scan order and subjected to selective scan operation, motivating architectures with composite inductive biases that relax the strict causality of selective scan operation at scale throughout the Deep Neural Network (DNN).

Prior works instrument the same conventional selective scan manifold (15), combined with different scan order permutations. One approach (Zhu *et al.*, [38]) proposes aggregation over the non-sequential input signal in different scan orders with different sets of SSM weights, followed by simple feature fusion. Another approach amortizes layer-wise heterogeneous scan order causal aggregations across depth of the DNN with no per-block parameter increase, as in *Zigma* [17], where each selective scan manifold $u \mapsto y$, captures dependencies between tokens at varying degrees of spatial vicinity (specified by the particular scan orders at corresponding layers) across layers in the DNN.

[1] We denote the terminal SSR as h_T; under zero-based indexing this is equivalently $h(T-1)$.

2.3.2. Generalizing the selective-scan manifold

Given the fact that the selective-scan operation aggregates information across input signal in a strictly causal manner, evolving the latent state-space representation (SSR) as Eq. (11), we hypothesize that the terminal SSR h_T computed during conventional selective scan Eq. (15) through a full causal pass over the non-sequential input signal, in specified scan order, potentially encodes a useful global summary and serves as a mild inductive cue for downstream state-space dynamics for subsequent blocks, despite the SSR being a severe compression of the non-sequential signal. We therefore generalize the conventional selective-scan manifold (15) with a Differentiable **R**ecurrent State **C**hain to *Arcee* selective-scan manifold (see Fig. 1(b)) that accepts initial value condition for internal state-space evolution dynamics and exposes it's terminal SSR h_T for upstream computation.

Arcee selective-scan manifold: We extend the conventional map (15) to a two-port block that *accepts* an initial state and *returns* its terminal state:

$$\mathcal{M}^{(\ell)}: \big(u^{(\ell)}(\cdot),\, h^{(\ell)}(0)\big) \longmapsto \big(y^{(\ell)}(\cdot),\, h^{(\ell)}(T)\big), \quad y^{(\ell)}(t) = \mathcal{F}\big(u^{(\ell)}(0{:}t)\big), \quad \forall\, t \in [0, T-1] \tag{16}$$

where the intra-block dynamics remain the causal selective scan of (12) (with ZOH factors $\bar{A}_t = e^{\Delta_t A}$, $\bar{B}_t = \big(\int_0^{\Delta_t} e^{\tau A} d\tau\big) B$).

Differentiable boundary map and cross block chaining: We connect Mamba blocks by a differentiable boundary map that seeds the next block's initial condition with the previous block's terminal SSR:

$$h^{(\ell)}(0) \;=\; \mathcal{T}^{(\ell)}\big(h^{(\ell-1)}(T-1)\big), \quad \mathcal{T}^{(\ell)} = \text{Identity by default.} \tag{17}$$

Even with such differentiable recurrent state chain, each block is still strictly causal internally and the $u^{(\ell)} \mapsto y^{(\ell)}$ mapping satisfies the lower-triangular Jacobian of the causal scan:

$$\frac{\partial y^{(\ell)}(j)}{\partial u^{(\ell)}(i)} = \begin{cases} D^{(\ell)}, & j = i, \\ C_j^{(\ell)}\Big(\prod_{k=i+1}^{j-1} \bar{A}_k^{(\ell)}\Big)\bar{B}_i^{(\ell)}, & i < j, \\ \mathbf{0}, & i > j, \end{cases}$$

Where empty product is I (identity).

Composing a DNN using L Mamba blocks with *Arcee* modification yields cross-depth system as follows:

$$\begin{aligned} (u^{(0)}, \ldots, u^{(L-1)}) &\longmapsto (y^{(0)}, \ldots, y^{(L-1)}), \\ h^{(\ell)}(0) = \mathcal{T}^{(\ell)}\big(h^{(\ell-1)}(T-1)\big),\ &\forall\, \ell \in \{1, \ldots, L-1\}, \\ h^{(0)}(0) &= 0 \quad \text{for block 0.} \end{aligned} \tag{18}$$

Cross-block Jacobian (Arcee). Because $h^{(\ell)}(0) = \mathcal{T}^{(\ell)}(h_T^{(\ell-1)})$, outputs of block ℓ depend on inputs of block $\ell-1$ only via the terminal handoff. Let $J_T^{(\ell)} = \partial h^{(\ell)}(0)/\partial h_T^{(\ell-1)}$ be the boundary Jacobian. Then for $j \in [0, T-1]$ and $i \in [0, T-1]$,

$$\frac{\partial\, y^{(\ell)}(j)}{\partial\, u^{(\ell-1)}(i)} = \underbrace{C_j^{(\ell)}\Bigg(\prod_{k=0}^{j-1} \bar{A}_k^{(\ell)}\Bigg)}_{\text{downstream readout through block } \ell} J_T^{(\ell)} \underbrace{\Bigg(\prod_{k=i+1}^{T-1} \bar{A}_k^{(\ell-1)}\Bigg) \bar{B}_i^{(\ell-1)}}_{\text{upstream contribution to } h_T^{(\ell-1)}}. \tag{19}$$

(Empty products equal the identity I.) Equation (19) consists of three parts: (i) an *upstream causal accumulator* that builds the terminal state $h_T^{(\ell-1)}$ inside block $\ell-1$; (ii) a *boundary map* $J_T^{(\ell)} = \partial h^{(\ell)}(0)/\partial h_T^{(\ell-1)}$ that hands off this terminal state to the next block (we default to $\mathcal{T}^{(\ell)}$=Identity); and (iii) a *downstream causal propagator* inside block ℓ that maps $h_0^{(\ell)}$ to outputs $y^{(\ell)}$. Therefore, each block stays strictly causal *inside* itself; only the compact terminal state is passed across blocks.

Network-level Jacobian and implicit limitations. If we stack blocks by depth, the overall Jacobian is *block–lower–triangular*: outputs of block ℓ never depend on inputs of any future block $m>\ell$. Any cross–block influence from block m to ℓ must pass through the terminal–state handoff $h_T \in \mathbb{R}^{d_{\text{inner}} \times d_{\text{state}}}$, which acts as a low–dimensional bottleneck. Consequently, the off–diagonal Jacobian blocks are low–rank; a safe, concise bound is

$$\operatorname{rank}\left(\frac{\partial y^{(\ell)}}{\partial u^{(m)}}\right) \;\le\; d_{\text{inner}} \cdot d_{\text{state}} \qquad \text{for } m < \ell. \tag{20}$$

i.e., cross–block effects are compressed by the SSR and cannot carry full per-token detail.

Equivalently, the implication is that each selective scan manifold throughout the DNN, starts with the terminal SSR h_T from previous block as a directional prior reflecting a full causal pass over input signal (in specified scan order). Although h_T is a severe compression, we hypothesize it still

encodes useful global summary. As a consequence, the differentiable recurrent state chain effectively acts as a architecture agnostic composite inductive bias across depth for Mamba based DNNs, in the sense that it can be plugged into selective-scan operation for any Mamba based DNN. Although the mamba blocks in isolation still remain causal, *Arcee* alleviates the strict causality of the selective-scan operation across the depth of the DNN, thereby increasing performance on non-sequential modalities as evident from our experiments discussed in results.

2.3.3. Implementation details (Arcee modifications)

We modify both the selective scan forward/backward fused kernels (introduced in [11]) to align with *Arcee selective-scan manifold* (16)

Forward (one read, one write):

$$h^{(\ell)}(0) \leftarrow \mathcal{T}^{(\ell)}\big(h^{(\ell-1)}(T-1)\big), \qquad \mathcal{T}^{(\ell)} = \mathrm{Id} \text{ by default.}$$

For the fused forward CUDA kernel, this adds a single read of $h^{(\ell-1)}(T-1)$ and a single write to $h^{(\ell)}(0)$.

Backward (seeded terminal adjoint): Let $g_{y^{(\ell)}(t)} = \partial\mathcal{L}/\partial y^{(\ell)}(t)$ and $g_{h^{(\ell)}(t)} = \partial\mathcal{L}/\partial h^{(\ell)}(t)$. The only change vs. vanilla is the *initial seed* for the terminal adjoint at block $\ell-1$ due to the boundary handoff into block ℓ:

$$g_{h^{(\ell-1)}(T-1)} \mathrel{+}= \underbrace{\frac{\partial\mathcal{L}}{\partial y^{(\ell-1)}}\frac{\partial y^{(\ell-1)}}{\partial h^{(\ell-1)}(T-1)}}_{\text{local (within block } \ell-1)} + \underbrace{\big(J_{\mathcal{T}}^{(\ell)}\big)^{\top} g_{h^{(\ell)}(0)}}_{\text{boundary from next block}}, \quad (21)$$

where $J_{\mathcal{T}}^{(\ell)} = \partial h^{(\ell)}(0)/\partial h^{(\ell-1)}(T-1)$. For the default identity mapping, $J_{\mathcal{T}}^{(\ell)} = I$ and the boundary term reduces to $g_{h^{(\ell)}(0)}$. Given the terminal seed (21), the per-token adjoint recurrences inside block ℓ are identical to conventional selective-scan:

$$\begin{aligned} g_{h^{(\ell)}(t)} &\mathrel{+}= (C_t^{(\ell)})^{\top} g_{y^{(\ell)}(t)} + (\bar{A}_t^{(\ell)})^{\top} g_{h^{(\ell)}(t+1)}, \\ g_{u^{(\ell)}(t)} &\mathrel{+}= (D_t^{(\ell)})^{\top} g_{y^{(\ell)}(t)} + (\bar{B}_t^{(\ell)})^{\top} g_{h^{(\ell)}(t+1)}. \end{aligned} \quad (22)$$

Parameter gradients accumulate as usual via chain rule through the selective heads: $(\bar{A}_t^{(\ell)}, \bar{B}_t^{(\ell)}, C_t^{(\ell)}, D_t^{(\ell)}, \Delta_t^{(\ell)}) = \Phi\big(u(t)^{(\ell)}; \theta\big)$.

Cost and memory. Arcee adds $O(d_{\text{state}})$ work per block from the boundary read/write and the terminal seed in (21). The scan FLOPs and memory traffic remain $O(T \cdot d_{\text{state}})$, which is identical to the vanilla fused selective scan. No additional activations are required beyond storing $h^{(\ell)}(T-1)$.

Stability note. With Hurwitz A and $0 < \Delta_{\min} \le \Delta_t \le \Delta_{\max}$, we have $\rho(e^{\Delta_t A}) < 1$ and the product $\prod_t e^{\Delta_t A}$ remains bounded, thus the SSM is stable for any h_0.

3. Experiments

3.1. Setup

We test the hypothesis that the terminal state–space representation h_T from a causal Mamba pass over a flattened, non-sequential signal acts as a mild *directional prior*: a compact global summary that, when reused across depth as initial value condition for state-space dynamics in subsequent block, improves downstream selective-scan dynamics yielding a composite inductive bias at scale that reduces friction when adapting Mamba blocks to non-sequential signals despite their strict causality in isolation.

Framework. We evaluate *Arcee* within the *Flow Matching* framework for unconditional image generation. We parameterize the *marginal vector field* $u_t(\cdot)$ (see Eq. (5)) with Mamba-based DNNs that process non-sequential signals (images in this case) as flattened tokens via selective scans.

Concretely, we compare Mamba-based DNNs that use the conventional selective scan (see Fig. 1(a), (15)) against the same DNNs where mamba blocks are augmented with the *Arcee* selective scan (a recurrent state chain that reuses terminal SSR as the initial condition for SSM dynamics in subsequent block through a differentiable boundary map across depth; see Fig. 1(b), (16)). To isolate the hypothesis under a fixed compute budget, we use a single dataset (CelebA-HQ 256×256) and omit unrelated backbones (e.g., Transformers/UNets [10, 30, 31]).

Deep Neural Network (DNN) Backbones. We integrate *Arcee* as a *boundary hook* that seeds each block's initial state $h^{(\ell)}(0)$ with the previous block's terminal SSR $h_T^{(\ell-1)}$ (Fig. 1(b), see Eq. 16), leaving the selective-scan body unchanged. This makes *Arcee* orthogonal to backbone specifics; we evaluate its effect on two Mamba-based DNN backbones: (i) **Zigzag-Mamba (Zigma)** [17], which amortizes heterogeneous scan-order permutations across depth so different Mamba blocks model relationships between tokens at varying spatial vicinities; (ii) **Vision Mamba** [38], which employs order-specific SSM weights followed by simple feature fusion.

We use Zigzag-Mamba (Zigma) and Vision Mamba in their original forms as published in [17, 38], without any architectural modification. Arcee is implemented purely as an extension of the conventional selective-scan manifold via a recurrent terminal-SSR chain and a differentiable boundary map across depth; no architectural diagrams change, only the $\big(h_T^{(\ell-1)}, h^{(\ell)}(0)\big)$ boundary is enabled (see Fig. 1).

Training details (Flow Matching). For the interpolation schedulers in Eq. (1) we use the generalized VP (GVP)

Table 1. Backbone specs (identical for baseline vs. +Arcee).

Family	depth	d_{model}	Params (M)
Zigma-k ($k \in \{1, 2, 4, 8\}$)	24	768	161.8
Vision Mamba	20	768	161.9

interpolant from SiT [28] with $\alpha_t = \cos\left(\frac{\pi}{2}t\right)$ and $\sigma_t = \sin\left(\frac{\pi}{2}t\right)$, sampling $t \sim \text{Uniform}[0, 1]$. Targets follow the conditional Flow Matching objective (Eq. 6). Unless noted otherwise, we train with AdamW (no weight decay), a constant learning rate of 3×10^{-4}, global batch size 192, image resolution 256^2, and 50,000 optimization steps; we enable RMSNorm, fused add–norm, and learnable positional embeddings, and set d_{state}=256. Budgets (steps, batch, sampler settings) are matched across each baseline and its +Arcee counterpart; Arcee uses $\mathcal{T} = \text{Id}$ and adds no extra parameters. All experiments are conducted in the latent space of a pre-trained variational autoencoder (VAE) [20, 33] with compression factor 8. We maintain an exponential moving average (EMA) of parameters with decay rate $\beta = 0.9999$ and report results using the EMA weights.

DNN configurations. For each DNN backbone we report depth L, embedding d_{model}, and parameter count. The state size is fixed to $d_{\text{state}=256}$ for all experiments. The +Arcee variants keep all hyperparameters fixed; only the (h_0, h_T) boundary is enabled (no extra params). To compensate for parameter overhead, per additional scan-order, due to order specific SSM weights, we reduce depth of the vision mamba backbone [38] to match parameter count with Zigma backbones.

Evaluation. We report **FID**↓ (CleanFID) on 50K samples and **KID**↓, both computed with Inception-V3 features [5, 29, 34]. Sampling uses the same ODE solver (Dormand–Prince, `dopri5`) and a fixed number of function evaluations (NFE = 50) across all models [9]. All models are trained for exactly 50,000 steps, and we evaluate the EMA checkpoint at this fixed step for every baseline and +Arcee variant.

3.2. Main results

Observation. On the naive single scan-order Zigma baseline, which is effectively strictly causal even across depth given the fact that all blocks employ the same scan-order to evaluate inherently selective-scan operation (15) over input signal, Arcee yields a large improvement: FID drops from 82.81 to 15.33 (5.4× lower) and KID from 88.69 to 10.59 as shown in Table 2, without changing the architecture or parameter count. For Zigma-4 (each block evaluates selective-scan over input in one of 4 different scan orders across depth; refer [17] for details), Arcee still provides a consistent gain (11.27 → 10.86 FID; 7.70 → 7.45 KID), indicating that cross-block SSR reuse remains beneficial even when some scan-order diversity is present. Fusion of selective-scan evolution over the input in multiple scan orders, each with its own SSM weights, already induces a strong inductive bias in Vision Mamba, making the causal selective scan much more amenable to non-sequential signals. As a result, Arcee yields a smaller but still consistent improvement (14.08 → 13.47 FID; 9.76 → 9.36 KID) on top of this architecture, suggesting that propagating the causal directional prior ($h^{(\ell)}(0) \leftarrow h_T^{(\ell-1)}$, refer Fig. 1(b), Eq. 16) can help across different Mamba-based DNNs.

Table 2. CelebA-HQ (256^2) Flow Matching. FID/KID: lower is better. All models trained for 50K steps with matched budgets; +Arcee variants add no parameters.

Model	FID↓	KID $\times10^3$ ↓
Zigma-1 (baseline)	82.81	88.69
+ Arcee (ours)	**15.33**	**10.59**
Zigma-4 (baseline)	11.27	7.70
+ Arcee (ours)	**10.86**	**7.45**
Vision Mamba (baseline)	14.08	9.76
+ Arcee (ours)	**13.47**	**9.36**

3.3. Baselines and ablations

Baselines. We consider two families of Mamba-based DNN backbones: (1) *Zigma-k* [17], a Zigzag-Mamba variant with k scan orders amortized across depth; and (2) *Vision Mamba* [38], which employs fusion of selective-scan evolution over input signal in multiple scan orders each with different sets of weights, thereby incurring parameter overhead for each additional scan-order. For each backbone, we train the baseline model and its +Arcee counterpart under identical budgets (Sec. 3.1).

Scan-order diversity (Zigma). To study the interaction between Arcee and scan-order diversity, we vary the number of Zigma scan orders $k \in \{1, 2, 4, 8\}$ and train each configuration with and without Arcee.

We observe three regimes (see Table 3). First, going from k=1 to k=4 improves the baseline, but pushing to k=8 already hurts performance: Zigma-8 is slightly worse than Zigma-4 under the same training budget, indicating diminishing and eventually negative returns from adding more scan orders to the baseline at fixed compute. Second, Arcee provides by far its largest gain in the low-diversity setting (k=1), is effectively neutral around k=2 (with small differences likely attributable to optimization noise), and gives a modest but consistent improvement at k=4. Third, in the heavily diversified setting k=8, Arcee-8 underperforms Zigma-8, suggesting that given the backbone's scan-order bias is already over-saturated and suboptimal, the additional directional prior cannot compensate

Table 3. Effect of scan-order diversity on Zigma with and without Arcee (CelebA-HQ 256^2). All models are trained for 50K steps with matched budgets.

Model	Orders k	FID↓	KID $\times 10^3$ ↓
Zigma-1	1	82.81	88.69
+ Arcee	1	**15.33**	**10.59**
Zigma-2	2	**15.64**	**11.22**
+ Arcee	2	15.68	11.40
Zigma-4	4	11.27	7.70
+ Arcee	4	**10.86**	**7.45**
Zigma-8	8	**11.19**	**7.77**
+ Arcee	8	11.75	8.17

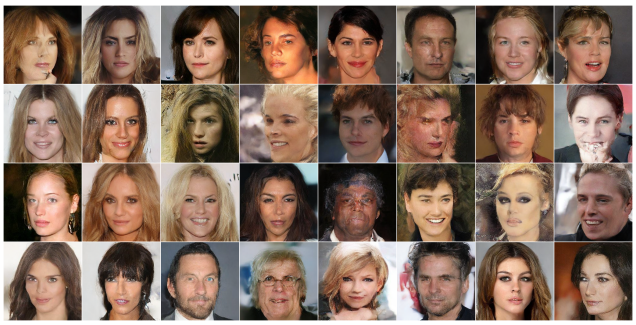

Figure 2. Zigma-1 + Arcee Qualitative CelebA-HQ (256^2) samples. Arcee enables sharper, more coherent faces under the same sampling budget.

and may mildly interfere with it.

Overall, the best-performing configuration in this family is Zigma-4+Arcee. This supports the view that Arcee is most useful in low- to moderate-diversity regimes, where cross-block SSR reuse complements existing architectural inductive biases by providing additional directional context across depth, while very aggressive scan-order diversification is evidently a poor design choice under our setup.

Cross-architecture generality (Vision Mamba). As shown in Tab. 2, Arcee also yields a smaller but consistent improvement on Vision Mamba, despite its strong built-in scan-order inductive bias via order-specific SSM weights. Together with the Zigma results above, this suggests that the proposed directional prior can be instrumented as a lightweight, plug-in boundary hook across different Mamba-based backbones, with the largest benefits appearing when global mixing is present but not already oversaturated by very aggressive scan-order diversification.

4. Conclusion

We developed *Arcee*, a structure-informed recurrent state chain for Mamba-based DNNs that reuses the terminal state–space representation h_T as a cross-block directional prior. Leveraging the underlying state–space dynamics, the design operates purely as a boundary hook on the selective-scan manifold, leaving the internal Mamba block, parameter count, and computational complexity unchanged. By propagating h_T across depth through a differentiable boundary map, Arcee induces a mild yet global inductive bias that helps adapt strictly causal selective scans to non-sequential signals.

Figure 3. Zigma-1 baseline Qualitative CelebA-HQ (256^2) samples.

Unlike prior "Mamba-for-vision" variants that rely primarily on scan order permutations or order-specific SSM weights, the proposed Arcee design is informed by the state–space structure itself: it explicitly reuses the terminal SSR as a compact global summary across blocks, while preserving per-block causality. Within the Flow Matching framework on CelebA-HQ (256^2), Arcee yields an 81.5% reduction in FID (82.81 $\rightarrow$ 15.33) and an 88.1% reduction in KID (88.69 $\rightarrow$ 10.59) on the naive single-order Zigma baseline, and provides consistent improvements for Zigma-4 ($\approx$ 3.6% FID and 3.2% KID gains) and Vision Mamba ($\approx$ 4.3% FID and 4.1% KID gains), all without adding parameters. Ablations over scan-order diversity further reveal that Arcee is most beneficial in low- to moderate-diversity regimes, while overly aggressive scan-ordering can itself become suboptimal under fixed budgets.

Future work. Arcee can be viewed as a generalization of the conventional selective-scan manifold to a two-port interface $(u, h^{(\ell)}(0)) \mapsto (u, h_T^{(\ell)})$: it supports nonzero, potentially learned initial value conditions for the state–space dynamics inside each selective scan, while exposing the terminal SSR h_T through a differentiable boundary map for upstream computation. This perspective suggests several directions, including conditioning selective-scan evolution in Mamba blocks on cross-modal priors (e.g., text, audio, or high-level semantic signals) via learned initial states, extending Arcee to conditional generative modeling and other modalities such as video and audio, and developing a sharper theoretical characterization of cross-block recurrence in continuous-time state–space models.

References

[1] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. JMLR version; v4. 1, 2

[2] V. I. Arnold. *Mathematical Methods of Classical Mechanics*. Springer, 2nd edition, 1989. Liouville theorem section. 2

[3] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22669–22679, 2023. 1

[4] Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *Proceedings of the 40th International Conference on Machine Learning*, pages 1692–1717. PMLR, 2023. 1

[5] Mikolaj Binkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR)*, 2018. 7

[6] Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. SE(3)-stochastic flow matching for protein backbone generation. *arXiv preprint arXiv:2310.02391*, 2024. ICLR 2024 Spotlight, v4 (Apr 11, 2024). 1

[7] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[8] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, pages 16344–16359. Curran Associates, Inc., 2022. 1

[9] J. R. Dormand and P. J. Prince. A family of embedded runge–kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980. 7

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 6

[11] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 1, 2, 3, 4, 6

[12] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. NeurIPS 2021. 1

[13] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR)*, 2022. ICLR 2022 (S4). 1

[14] Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. NeurIPS 2022. 1

[15] Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Ré. How to train your hippo: State space models with generalized orthogonal basis projections. *arXiv preprint arXiv:2206.12037*, 2022. 1

[16] Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. NeurIPS 2022. 1

[17] Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes Fischer, and Björn Ommer. Zigma: A dit-style zigzag mamba diffusion model. In *Computer Vision – ECCV 2024*, pages 148–166. Springer, 2024. 2, 4, 6, 7

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 2

[19] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

[20] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 2014. 7

[21] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, Matt Le, et al. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. v2. 1, 2

[22] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024. 2

[23] Jiarun Liu, Hao Yang, Hong-Yu Zhou, Yan Xi, Lequan Yu, Yizhou Yu, Yong Liang, Guangming Shi, Shaoting Zhang, Hairong Zheng, and Shanshan Wang. Swin-umamba: Mamba-based unet with imagenet-based pretraining. *arXiv preprint arXiv:2402.03302*, 2024. 1

[24] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 1, 2

[25] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1

[26] Cheng Lu, Yu Lu, Jianfei He, Hangjie Ren, Jinghao Wang, Fan Bao, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[27] Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024. 1

[28] N. Ma, M. Goldstein, Michael S. Albergo, N. M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and

diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision (ECCV)*, pages 23–40. Springer, 2024. 7

[29] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in FID calculation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11410–11420, 2022. 7

[30] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2023. 1, 6

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015. 6

[32] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1

[33] Stability AI. Improved VAE for Stable Diffusion (sd-vae-ft-ema). https://huggingface.co/stabilityai/sd-vae-ft-ema, 2022. 7

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. 7

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. 1

[36] J. Wang, T. Gangavarapu, J. N. Yan, and A. M. Rush. Mambabyte: Token-free selective state space model. *arXiv preprint*, 2024. arXiv preprint. 4

[37] S. Wang and B. Xue. State-space models with layer-wise nonlinearity are universal approximators with exponentially decaying memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 4

[38] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Proceedings of the 41st International Conference on Machine Learning*, pages 62429–62442. PMLR, 2024. 1, 4, 6, 7