Adaptive Filter Convolution

Peter Racioppo
pcracioppo@gmail.com

***Abstract*—Convolutional Neural Networks (CNNs) rely on discrete kernels that treat spatial aggregation as a generic weighted sum, lacking an explicit model of the underlying continuous spatial dynamics. We present Adaptive Filter Convolution, a continuous-time framework that replaces standard static kernels with a learnable, precision-weighted filter derived from 2D Stochastic Differential Equations (SDEs). By modeling the image latent state as a Matérn Gaussian Random Field governed by decoupled axial Ornstein-Uhlenbeck processes, we derive a closed-form Maximum Likelihood estimator that functions as a locally parallelized Kalman filter. This formulation injects principled inductive biases—specifically spatial smoothness, exponential decay, and complex rotational dynamics—directly into the kernel structure while maintaining the $\mathcal{O}(N)$ complexity of standard convolution. Unlike standard attention, which computes similarity in an unstructured space, our mechanism computes a robust Mahalanobis distance in the precision-weighted eigenbasis of the dynamics. We demonstrate that this structurally constrained kernel yields a more parameter-efficient and geometrically interpretable alternative to standard convolutions, providing a rigorous probabilistic unification of spatial filtering and deep feature extraction.**

***Index Terms*—Stochastic Differential Equations, Attention Convolution, Dynamic Convolution, Adaptive Filtering**

I. Introduction

A. The AFA Convolutional Kernel

Our core innovation is embedding the mathematics of Stochastic Differential Equations (SDEs)—specifically, the Ornstein-Uhlenbeck (OU) process, which underlies the Matérn class of Gaussian Random Fields (GRFs)—directly into the convolution layer structure. This provides key advantages over existing local attention methods: The kernel's transition matrix is constrained by the analytic form of the OU process's solution (exponential decay and complex rotation), which efficiently enforces stability, spatial smoothness, and rotational dynamics. The output is computed as a precision-weighted Maximum Likelihood Estimate (MLE), effectively a local, parallelized Kalman filter step. Because the operation is confined to the kernel neighborhood, AFA convolution preserves the asymptotic complexity of standard attention-convolution layers while adding principled uncertainty weighting and dynamical structure.

This is a direct application of the Adaptive Filter Attention (AFA) mechanism developed in *Attention as an Adaptive Filter* [1], adapted here for computer vision. In this work, attention is derived as the optimal update rule of an adaptive filter for a latent trajectory governed by a stable linear SDE. With additional constraints—namely that the query–key dynamics have isotropic decay and noise—the formulation reduces to a scalable, complex-valued generalization of dot-product attention with Rotary Positional Embeddings (RoPE).

II. Related Work

A. Background: SPDEs and Optimal Feature Fusion

Stochastic Partial Differential Equations (SPDEs) offer a mathematically rigorous way to model image date: the solution to a linear SPDE driven by white noise is a Gaussian Random Field (GRF), $\mathbf{x}(\mathbf{s})$. This framework defines pixel correlation as arising from fundamental laws like diffusion and stability.

The goal of optimally filtering image data requires computing the precision matrix, $\mathbf{Q} = \mathbf{\Sigma}^{-1}$, which is computationally intractable ($\mathcal{O}(N^3)$) for a dense covariance matrix $\mathbf{\Sigma}$. To overcome this fundamental problem, Lindgren, Rue, and Lindström established that Matérn Gaussian Fields can be explicitly approximated by discretely-indexed Gaussian Markov Random Fields [2]. The underlying local SPDE dynamics guarantee that the inverse covariance (the precision matrix $\mathbf{Q}$) is sparse, which decreases the inference cost to ($\mathcal{O}(N^{3/2})$ for a 2D field). Following this, the Deep GMRF (DGMRF) framework formalized the equivalence between GMRFs and CNNs, using a multi-layer CNN to define the sparse precision structure and achieving linear-time scalability for probabilistic inference in spatial modeling [3].

Our approach extends this line of work by embedding these principles directly into convolutional kernels via a separable SDE formulation. We use the simplest, most tractable case: the Ornstein–Uhlenbeck (OU) process. The OU process is characterized by the exponential covariance kernel, which is the analytic solution to a stable linear SDE:

$$C(d) = \sigma^2 \exp\left(-\frac{d}{\rho}\right)$$

By using decoupled linear SDEs along the x and y axes, we achieve a compact and interpretable prior for spatial dynamics. The layer output is a precision-weighted sum of neighborhood evidence, a Kalman filter-like update applied in parallel across all neighbors. This structure yields a compact kernel that encodes a principled probabilistic prior for spatial features.

B. Structural Priors in Convolution

Standard CNNs require large datasets to implicitly capture basic geometric priors such as rotation and smoothness. Enforcing analytic constraints makes local features more compact and interpretable, reducing parameters and encouraging the network to learn general transformation laws rather than arbitrary weights—a key advantage in low-data regimes.

This principle is realized through several methods that enforce explicit constraints on the filter space. Group-equivariant CNNs (G-CNNs) generalize standard convolutions to arbitrary transformation groups, enabling built-in rotational and reflectional equivariance [4]. Steerable convolutions further parameterize filters as linear combinations of steerable basis functions, ensuring continuous rotation equivariance and predictable feature-map transformations under planar symmetries [5]. GaborNet replaces fixed kernels with learnable Gabor filters to capture frequency-localized features [6], while Um et al. show that 2D oscillatory kernels can be factorized into separable axial operations for efficient computation [7].

Several works augment local aggregation with explicit relative positional information. Local Relation Networks (LRN) [8] compute content-dependent weights over a local patch that are explicitly modulated by the relative spatial offsets of the pixels $(\Delta x, \Delta y)$. Similarly, Attention-Augmented Convolutions [9] often incorporate relative positional embeddings into the scoring function, providing a learned bias based on the spatial difference between tokens.

In our method, the state transition matrix $\mathbf{\Phi}$ analytically embeds the principle of stable dynamics—exponential decay and complex rotation—directly into the kernel weights.

C. Neural Networks for State Estimation

Several works reinterpret convolutional layers through the lens of dynamical systems and state estimation. Gramlich et al. [10] analyze CNNs as two-dimensional linear shift-invariant systems, deriving their impulse and frequency responses and showing how convolution corresponds to a structured 2-D system representation.

Deep Kalman Filters (DKFs) extend classical Kalman filtering to high-dimensional nonlinear dynamics by using neural networks to parameterize transition and observation models, while maintaining a probabilistic latent state with explicit uncertainty propagation [11]. More recent approaches replace or augment the Kalman gain with learned modules: KalmanNet uses a recurrent network to adaptively compute the gain [12], while the Attention Kalman Filter (AtKF) and KalmanFormer substitute the gain with attention mechanisms or Transformer blocks [13, 14]. These works reflect a broader trend of using neural networks for adaptive state estimation, but most still augment the sequential structure of the Kalman filter. By contrast, our method embeds a closed-form version of the Kalman correction step directly into a neural layer, allowing it to be applied in parallel.

D. Attention and State Space Models for Vision

Modern attention architectures require highly structured positional embeddings such as rotary positional embeddings (RoPE) [15] to efficiently encode spatial relationships. EVA-02 introduced 2D RoPE to encode relative spatial information in its Vision Transformer architecture [16]. Heo et al. generalize 2D RoPE to RoPE-Mixed, which applies a learned mixed-frequency 2D rotation to queries and keys [17]. CoPE [18] and ComplexFormer [19] extend RoPE into the complex domain.

Our SDE kernel builds on this trend but provides a complete, principled filter structure. Unlike RoPE, which only borrows complex rotation geometry for positional encoding within a real-valued attention mechanism, we embed the structural dynamics of the stable SDE into an end-to-end complex-valued filter. This allows for learnable coupling between real and imaginary components and leverages the full complex similarity metric, ensuring mathematical fidelity to the underlying dynamical system.

State space models provide an alternative approach for efficiently capturing long-range dependencies. The Structured State Space Sequence model (S4) achieves linear-time sequence modeling by representing system matrices in a diagonal-plus-low-rank form [20]. Gu et al. introduce Linear State-Space Layers (LSSLs), which unify recurrent, convolutional, and continuous-time approaches through structured linear dynamics [21]. S4ND generalizes 1D state space models (SSMs) to multidimensional signals, enabling images and videos to be modeled using continuous, global convolutional kernels. By factoring these kernels as low-rank tensor products of 1D SSMs, S4ND achieves parameter efficiency, long-range context, and resolution invariance [22]. Baron et al. introduce a 2-D State Space Model (2D-SSM), which models an image's latent features using coupled horizontal and vertical recurrent states, updating a combined state vector based on the input and neighboring states via four learned transition matrices. The key insight is that this recurrence is linear and can be unrolled into a non-local 2D convolution kernel [23]. Another recent line of work frames convolutional layers as multidimensional linear dynamical systems using Roesser-type state space models, providing minimal, compact representations [24].

E. Attention-Convolutional Hybrids

The idea of performing local attention within convolutional kernels has been explored in various forms. Dynamic Filter Networks [25] and subsequent dynamic convolution variants, such as CondConv [26], replace fixed convolution weights with input-conditioned local kernels, yielding spatially varying, feature-dependent filters. Local and windowed self-attention mechanisms have been widely adopted, including Stand-Alone Self-Attention [27] and the Neighborhood Attention Transformer [28], which constrain the query to a local $K \times K$ neighborhood, mirroring convolution's receptive field. The expressive equivalence between convolution and self-attention was established by Cordonnier et al. [29], framing attention as a content-adaptive generalization. Related efforts like Involution [30] generate per-location kernels that modulate values element-wise.

Depthwise convolution combined with local attention mechanisms has been explored in various works. Han et al. [31] demonstrated that local self-attention can be reframed as a channel-wise, locally connected layer closely resembling dynamic depthwise convolution , with comparable or superior empirical performance. CvT [32] adopts a hybrid approach where the query, key, and value projections in the Transformer are replaced by depthwise-separable convolutions, embedding

local receptive-field structure directly into the attention mechanism. Large Kernel Attention [33, 34] constructs spatial attention masks using stacked and dilated depthwise convolutions rather than dot-product attention, effectively realizing "attention inside the kernel" over large neighborhoods.

Other work explores feature-wise gating or non-Euclidean scoring. The Gated Attention Unit [35] replaces dot-products with Hadamard gating of Q and K. FiLM [36] applies conditional per-feature scaling. Elliptical Attention [37] replaces Euclidean similarity with a learnable Mahalanobis metric, conceptually aligning with the necessity of precision-weighted scoring, though not tied to per-offset, diagonal kernels.

This prior work establishes three separate lines of precedent: local center-to-neighborhood attention, per-offset feature modulation, and robust Mahalanobis-like similarity. Yet, to our knowledge, no prior method combines these with an explicit structural derivation: namely, a local attention block featuring complex rotational dynamics $\mathbf{\Phi}$, per-offset Hadamard gating of values, and an explicit diagonal precision $\mathbf{p}^C$.

III. Methods

A. AFA Convolutional Kernels

Our goal is a convolutional filter structure that yields a closed-form, precision-weighted optimal estimate while ensuring stability and linear-time scalability ($\mathcal{O}(N)$). We achieve this by defining the local feature relationships using the simplest set of analytical dynamics: decoupled, stable linear SDEs (Ornstein–Uhlenbeck (OU) processes). This choice satisfies our core requirements: (1) linearity ensures a closed-form solution for the covariance propagation via the differential Lypuanov equation (DLE), and (2) stability ensures convergence to a stationary, short-range correlation structure. This OU-process formulation is statistically desirable because its exponential covariance aligns precisely with the Matérn Gaussian Field with smoothness parameter $\nu = \frac{1}{2}$, providing a principled, first-order probabilistic interpretation of our feature dynamics.

In particular, for a state $\mathbf{x}_{ij} \in \mathbb{R}^d$ at grid position (i, j), we define two directional SDEs radiating into position (i, j) from the horizontal and vertical directions:

$$d\mathbf{x}^h_{ij} = \mathbf{A}_h \mathbf{x}^h_{ij}\, di + \mathbf{G}_h\, d\mathbf{w}^{(h)}, \quad d\mathbf{x}^v_{ij} = \mathbf{A}_v \mathbf{x}^v_{ij}\, dj + \mathbf{G}_v\, d\mathbf{w}^{(v)},$$

$$d\mathbf{w}^{(h)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}|di|), \quad d\mathbf{w}^{(v)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}|dj|), \tag{1}$$

where $d\mathbf{w}^{(h)}$, $d\mathbf{w}^{(v)}$ are each standard Wiener increments. The process noise covariance for each SDE is $\mathbf{Q}_\ell := \mathbf{G}_\ell \mathbf{G}_\ell^\top$, $\ell \in \{h, v\}$. We simplify the SPDE problem to two decoupled axial SPDEs because this is the minimal and most tractable set required to capture the essential properties of a 2D filter: non-causality, separability, and the full directional Markov property. We assume noisy measurements of the form:

$$\mathbf{z}_{ij} = \mathbf{x}_{ij} + \mathbf{v}_{ij}, \quad \mathbf{v}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \tag{2}$$

We include a coordinate transformation to allow the SDEs to be oriented along arbitrary directions: Given two measurements $\mathbf{z}_{ij}, \mathbf{z}_{kl} \in \mathbb{R}^d$, let $\Delta x = i - k$, $\Delta y = j - l$. Let $\mathbf{T} \in \mathbb{R}^{2\times 2}$, $\mathbf{b} \in \mathbb{R}^{2\times 1}$ and define:

$$\begin{pmatrix} \Delta x' \\ \Delta y' \end{pmatrix} = \mathbf{T} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} + \mathbf{b}, \quad \Delta x = i - k,\ \Delta y = j - l. \tag{3}$$

This transformation allows the filter to model learnable anisotropy—a key property of Matérn fields—while preserving the separability required for the closed-form solution of the DLE.

We wish to estimate $\{\mathbf{x}_{ij}\}$ from the corrupted measurements $\{\mathbf{z}_{ij}\}$, for all $i, j \in \mathbb{R}^{H\times W}$. Given a measurement $\mathbf{z}_{kl}$, we can form an estimate of $\mathbf{x}_{ij}$ from m neighbors in a neighborhood $\mathcal{N}$ as:

$$\hat{\mathbf{z}}_{kl,ij} := \mathbf{\Phi}_{kl,ij} \mathbf{z}_{kl} \tag{4}$$

where $\mathbf{\Phi}_{kl,ij}$ is a state transition matrix mapping a state through the dynamics from position kl to ij. The maximum likelihood estimate of $\mathbf{x}_{ij}$, given a set of measurements $\mathbf{z}_{kl}$ in a neighborhood $\mathcal{N}$, is given by a precision-weighted sum:

$$\bar{\mathbf{z}}_{ij} = \Big(\sum_{u,v\in\mathcal{N}} \mathbf{P}^C_{uv} \Big)^{-1} \sum_{k,l\in\mathcal{N}} \mathbf{P}^C_{uv}\, \hat{\mathbf{z}}_{(i,j),(i+u,j+v)}, \tag{5}$$

where the total precision $\mathbf{P}^C_{uv}$ requires solving the differential Lyapunov equation (DLE) along each axis ℓ:

$$\frac{d}{d\tau}\mathbf{V}_\ell(\tau) = \mathbf{A}_\ell \mathbf{V}_\ell(\tau) + \mathbf{V}_\ell(\tau)\mathbf{A}_\ell^\top + \mathbf{Q}_\ell, \tag{6}$$

In general, solving the DLE for the matrix $\mathbf{V}_\ell$ and inverting the resulting $d \times d$ precision matrix $\mathbf{P}^C$ scales as $\mathcal{O}(d^3)$ per pixel pair. To achieve a closed-form, parallelized solution, we assume that all system and noise matrices are simultaneously diagonalizable by an invertible matrix $\mathbf{S} \in \mathbb{C}^{d\times d}$:

$$\mathbf{A}_\ell = \mathbf{S}\mathbf{\Lambda}_\ell \mathbf{S}^{-1},\ \mathbf{Q}_\ell = \mathbf{S}\mathbf{\Lambda}_{Q,\ell}\mathbf{S}^\dagger,\ \mathbf{R} = \mathbf{S}\mathbf{\Lambda}_R\mathbf{S}^\dagger, \tag{7}$$

where $\mathbf{\Lambda}_\ell, \mathbf{\Lambda}_{Q,\ell}, \mathbf{\Lambda}_R \in \mathbb{R}^{d\times d}$ are diagonal. This decouples the vector-valued filtering problem into d independent scalar problems in the eigenbasis. Let:

$$\mathbf{\Lambda}_\ell = -\text{diag}(\boldsymbol{\alpha}_\ell^2) + i\text{diag}(\boldsymbol{\omega}_\ell) \tag{8}$$

where $\boldsymbol{\alpha}_\ell^2$ is a vector of learned decay rates, which is squared and negated to ensure the eigenvalues are negative (i.e., $\text{Re}(\mathbf{\Lambda}_\ell) \le \mathbf{0}$), which guarentees the stability of the underlying OU process. We define the noise parameters as:

$$\mathbf{\Lambda}_{Q,\ell} = \text{diag}(\boldsymbol{\sigma}_\ell^2),\ \mathbf{\Lambda}_R = \text{diag}(\boldsymbol{\eta}^2), \tag{9}$$

where the parameters are squared to enforce positivity.

With these assumptions, we can map our measurements into the eigenbasis using $\mathbf{z}_{\mathbf{s},ij} := \mathbf{S}^{-1}\mathbf{z}_{ij}$. Since $e^{\mathbf{A}_\ell} = \mathbf{S}e^{\mathbf{\Lambda}_\ell}\mathbf{S}^{-1}$, we can define a state transition matrix in the eigenbasis as:

$$\begin{aligned} \mathbf{\Phi}_x(\Delta x') &= e^{-\boldsymbol{\alpha}_x^2|\Delta x'|}\, e^{-i\boldsymbol{\omega}_x \Delta x'}, \\ \mathbf{\Phi}_y(\Delta y') &= e^{-\boldsymbol{\alpha}_y^2|\Delta y'|}\, e^{-i\boldsymbol{\omega}_y \Delta y'}, \\ \mathbf{\Phi}_{xy}(\Delta x', \Delta y') &= \mathbf{\Phi}_x(\Delta x') \odot \mathbf{\Phi}_y(\Delta y') \in \mathbb{C}^d. \end{aligned} \tag{10}$$

Our estimates are then:

$$\hat{\mathbf{z}}_{\mathbf{s},(ij),(kl)} := \mathbf{\Phi}_{xy}(\Delta x', \Delta y') \odot \mathbf{z}_{\mathbf{s},(kl)}. \tag{11}$$

The DLE decouples into d scalar differential equations in the eigenbasis, which can be solved in closed form. For a decoupled random field in 2D, the state covariance is a product of the 1D covariances propagated in each direction. Hence, the total covariance in the eigenbasis is:

$$\begin{aligned}\boldsymbol{\sigma}_V^2(\Delta x', \Delta y') = \left(\boldsymbol{\sigma}_x^2 \frac{e^{-2\boldsymbol{\alpha}_x^2|\Delta x'|} - \mathbf{1}}{-2\boldsymbol{\alpha}_x^2}\right)\left(\boldsymbol{\sigma}_y^2 \frac{e^{-2\boldsymbol{\alpha}_y^2|\Delta y'|} - \mathbf{1}}{-2\boldsymbol{\alpha}_y^2}\right) \\ + \boldsymbol{\eta}^2 e^{-2\boldsymbol{\alpha}_x^2|\Delta x'| - 2\boldsymbol{\alpha}_y^2|\Delta y'|} + \boldsymbol{\gamma}^2,\end{aligned} \tag{12}$$

where the first term is accumulated process noise and the second term is measurement noise propagated into the anchor frame (and where division indicates element-wise division). The stabilization term $\boldsymbol{\gamma}^2$ is added to the covariance to model the static, intrinsic noise floor of the current anchor measurement $\mathbf{z}_i$ and to allow the model to decouple the final estimate's uncertainty from the strict SDE dynamics, treating the MLE at each pixel as an independent, local optimization. The total propagated precision is then $\mathbf{P}_{uv}^C = \mathbf{S}\text{diag}(\mathbf{p}_{uv})\mathbf{S}^\dagger$, where:

$$\mathbf{p}_{uv} = 1/\boldsymbol{\sigma}_V^2(\Delta x', \Delta y') \in \mathbb{R}^{d \times m} \tag{13}$$

We can include an attention-like mechanism between the central feature vector and all other feature vectors in the kernel by defining residuals $\mathbf{r}_{(ij),(kl)} = \mathbf{z}_{\mathbf{s},(ij)} - \hat{\mathbf{z}}_{\mathbf{s},(ij),(kl)}$ and using a residual-based reweighting of the precisions:

$$\mathbf{p}_{kl}^C = \frac{\mathbf{p}_{kl}}{1 + \mathbf{r}_{(ij),(kl)}^\dagger(\mathbf{p}_{kl} \odot \mathbf{r}_{(ij),(kl)})} \tag{14}$$

(of the form commonly found in iteratively-reweighted least squares (IRLS) or Robust Kalman Filtering.)

For a neighborhood $\mathcal{N}$, we can now define a convolutional kernel:

$$\bar{\mathbf{z}}_{\mathbf{s},ij} = \left(\sum_{(u,v)\in\mathcal{N}} \mathbf{p}_{uv}^C\right)^{-1} \odot \sum_{(u,v)\in\mathcal{N}} \mathbf{p}_{uv}^C \odot \hat{\mathbf{z}}_{\mathbf{s},(ij),(i+u,j+v)}. \tag{15}$$

Hence, we down-weight neighbors whose propagated predictions disagree strongly with the anchor.

Note that this structure is very similar to the Gabor Filter, but with two oscillation terms, vector-valued frequency parameters $\boldsymbol{\omega}_\ell$, and a Laplacian rather than Gaussian envelope.

Finally, we map back to the original basis: $\bar{\mathbf{z}}_{ij} = \mathbf{S}\bar{\mathbf{z}}_{\mathbf{s},ij}$.

B. Computational Advantages

The simultaneous diagonalization assumption (Eq. (7)) is the structural key that minimizes the cost of computing the final fusion weights for a single anchor pixel.The diagonalization converts the potentially demanding $\mathcal{O}(md^3)$ cost (required to compute the DLE propagations and matrix inversions for m neighboring features) into a parallelized cost of $\mathcal{O}(md)$ per anchor pixel. This cost reduction is achieved because the precision matrix in the eigenbasis, $\mathbf{P}^C$, is now a diagonal vector ($\mathbf{p}$), eliminating all costly matrix multiplication/inversion steps.

While global Transformer attention (where the sequence length m may be large) must simplify the complex Mahalanobis distance to a scalar $\mathbf{r}^\top\mathbf{r}$ term, our local design avoids this. Because the AFA Convolutional Kernel has only a single query per kernel window (one anchor pixel), we can afford the full vectorized Mahalanobis distance calculation, $\mathbf{r}_{(ij),(kl)}^\dagger(\mathbf{p}_{kl} \odot \mathbf{r}_{(ij),(kl)})$, across the feature dimension d, with a precision $\mathbf{p}_{kl}^C$ that can vary across the feature channels at each offset, providing superior feature-wise control over scalar attention mechanisms. For the same reason, the expansion and factorization tricks used in AFA become unnecessary.

As in standard convolution, the fixed, small size of the convolutional kernel ($m = K^2$) ensures the total complexity remains viable within a deep architecture.

C. Implementation

The SDE kernel's complex-valued operations can be executed entirely in the real domain. We define separate query, key, value, and output mappings using 1×1 convolutions. The real-valued input features $\mathbf{Z} \in \mathbb{R}^{d \times m}$ are projected into the complex eigenbasis, forming $2d$-dimensional query, key, and value vectors $\mathbf{Z_q}, \mathbf{Z_k}, \mathbf{Z_v} \in \mathbb{R}^{2d \times m}$:

$$\mathbf{Z}_\ell = \begin{bmatrix}\mathbf{Z}_{\ell r} \\ \mathbf{Z}_{\ell i}\end{bmatrix} = \mathcal{L}_\ell^{\mathbb{R}\to\mathbb{C}}(\mathbf{Z}) = \begin{bmatrix}\mathbf{W}_{\ell r} \\ \mathbf{W}_{\ell i}\end{bmatrix}\mathbf{Z} + \begin{bmatrix}\mathbf{b}_{\ell r} \\ \mathbf{b}_{\ell i}\end{bmatrix} \tag{16}$$

for $\ell \in \{\mathbf{q}, \mathbf{k}, \mathbf{v}\}$.

In the complex eigenbasis, each feature dimension evolves independently, so all operations reduce to fast, parallel, element-wise multiplications and summations over the real and imaginary parts. The core operation is the propagation of the neighbor values and keys $\mathbf{Z_v}, \mathbf{Z_k}$ within each kernel to the frame of the query $\mathbf{z_q}$ via the transition matrix $\mathbf{\Phi}_{xy}$. Because of the simultaneous diagonalizability assumption, this is a complex element-wise multiplication, where $\mathbf{\Phi}_{xy}$ dictates the rotation (phase correction) and scaling (decay). Complex multiplication adds the phases, so we can define cosine, sine, and exponential decay matrices:

$$\begin{aligned}\mathbf{C}(\Delta x', \Delta y') &:= \cos\left(\boldsymbol{\omega}_x \Delta x' + \boldsymbol{\omega}_y \Delta y'\right), \\ \mathbf{S}(\Delta x', \Delta y') &:= \sin\left(\boldsymbol{\omega}_x \Delta x' + \boldsymbol{\omega}_y \Delta y'\right), \\ \mathbf{E}(\Delta x', \Delta y') &:= e^{-\boldsymbol{\alpha}_x^2|\Delta x'| - \boldsymbol{\alpha}_y^2|\Delta y'|}\end{aligned} \tag{17}$$

Our state transition matrix is then:

$$\begin{aligned}\mathbf{\Phi}_{xy}^{\text{batch}} &= \mathbf{\Phi}_r^{\text{batch}} + i\,\mathbf{\Phi}_i^{\text{batch}} \in \mathbb{C}^{d \times m}, \\ \mathbf{\Phi}_r^{\text{batch}} &= \mathbf{E} \odot \mathbf{C}, \quad \mathbf{\Phi}_i^{\text{batch}} = -\mathbf{E} \odot \mathbf{S}.\end{aligned} \tag{18}$$

We next map the neighbor states to the anchor's coordinate frame, providing the estimates $\hat{\mathbf{Z}}_\mathbf{v}, \hat{\mathbf{Z}}_\mathbf{k}$.

$$\begin{aligned}\hat{\mathbf{Z}}_{\ell r} &= (\mathbf{\Phi}_r^{\text{batch}} \odot \mathbf{Z}_{\ell r}) - (\mathbf{\Phi}_i^{\text{batch}} \odot \mathbf{Z}_{\ell i}), \\ \hat{\mathbf{Z}}_{\ell i} &= (\mathbf{\Phi}_r^{\text{batch}} \odot \mathbf{Z}_{\ell i}) + (\mathbf{\Phi}_i^{\text{batch}} \odot \mathbf{Z}_{\ell r}),\end{aligned} \tag{19}$$

for $\ell \in \{\mathbf{k}, \mathbf{v}\}$.

First, the residual tensor $\mathbf{R} \in \mathbb{C}^{d\times m}$ is calculated by broadcasting the single anchor query across the m dimension:

$$\mathbf{R}_{(ij),(kl)} = \mathbf{z}_{\mathbf{q},(ij)}\mathbf{1}_m^\top - \hat{\mathbf{Z}}_{\mathbf{k},(ij),(kl)} \tag{20}$$

The element-wise square of the Mahalanobis distance ($\mathbf{D}_{\text{Mahal}}$) for all $d \times m$ elements is then computed using Hadamard products:

$$\mathbf{D}_{\text{Mahal}} = \mathbf{1}_{1\times d}\left(\mathbf{R}^\dagger \odot (\mathbf{P} \odot \mathbf{R})\right) \in \mathbb{R}^{1\times m} \tag{21}$$

where $\mathbf{1}_{1\times d}$ is a vector of ones, $\mathbf{P}^C \in \mathbb{R}^{d\times m}$ is the tensor of base precisions and $\mathbf{R}^\dagger$ denotes element-wise complex conjugation.

The final reweighted precision tensor, $\mathbf{P}^C$, is calculated via element-wise division ($\oslash$). This step applies the robust factor (the Mahalanobis distance term) to the base precision across all features and offsets simultaneously:

$$\mathbf{P}^C = \mathbf{P} \oslash (\mathbf{1}_{1\times m} + \mathbf{D}_{\text{Mahal}}) \tag{22}$$

where $\mathbf{1}_{1\times m}$ is a vector of ones. This equation is implemented by broadcasting the $1 \times m$ denominator across the d feature dimension to match the shape of the $d \times m$ base precision $\mathbf{P}$.

The final MLE numerator ($\mathbf{\Sigma}_F$) is the summation of contributions across all m neighbors. This is executed as a vectorized operation on the tensors, followed by a reduction sum along the m dimension. Let $\hat{\mathbf{Z}}_\mathbf{v} \in \mathbb{C}^{d\times m}$ be the tensor of propagated values, and $\mathbf{M} \in \mathbb{R}^{d\times m}$ be the broadcasted boundary mask. The numerator is:

$$\mathbf{\Sigma}_F = \sum_{m\in\mathcal{N}} \left[\mathbf{P}^C \odot \hat{\mathbf{Z}}_\mathbf{v} \odot \mathbf{M}\right] \in \mathbb{C}^d \tag{23}$$

where $\sum_{m\in\mathcal{N}}$ denotes summation along the neighbor dimension. The normalization term is the sum of the valid precision weights:

$$\mathbf{\Sigma}_P = \sum_{m\in\mathcal{N}} \left[\mathbf{P}^C \odot \mathbf{M}\right] \in \mathbb{R}^d \tag{24}$$

The final precision-weighted average is the vectorized division:

$$\bar{\mathbf{Z}}_\mathbf{v} = \mathbf{\Sigma}_F \oslash \mathbf{\Sigma}_P \tag{25}$$

This defines the output of a single convolution; repeating this calculation across all N anchor pixels defines the convolutional layer. The $2d$-dimensional result is projected back to the real domain by a final 1×1 convolution, combining the real and inverted imaginary components to produce the final $d \times m$ dimensional output of the AFA convolutional layer:

$$\bar{\mathbf{Z}} := \mathcal{L}_\mathbf{p}^{\mathbb{C}\to\mathbb{R}}(\bar{\mathbf{Z}}_\mathbf{v}) = (\bar{\mathbf{Z}}_{\mathbf{v}r}\mathbf{W}_{\mathbf{p}r}^\top) - (\bar{\mathbf{Z}}_{\mathbf{v}i}\mathbf{W}_{\mathbf{p}i}^\top) + \mathbf{b}_{\mathbf{p}r} \tag{26}$$

Finally, we include a residual connection for stability:

$$\bar{\mathbf{Z}} \leftarrow \bar{\mathbf{Z}} + \mathbf{Z} \tag{27}$$

Conceptually, the complex rotation operation is a generalization of Rotary Positional Embeddings (RoPE), where the phase rotation is defined by the analytically-derived SDE dynamics and not a fixed sinusoidal function.

Alternatively, we can let either or both of $\mathbf{P}^C \in \mathbb{R}^{d\times m}$ and $\mathbf{\Phi}_{xy}^{\text{batch}} \in \mathbb{R}^{2d\times m}$ be entirely learned, which trades our strong inductive bias for greater expressivity. We explore three variants:
(1) Analytic $\mathbf{P}^C$ and $\mathbf{\Phi}_{xy}^{\text{batch}}$, parameterized by $\{\sigma_x, \sigma_y, \eta, \gamma, \alpha_x, \alpha_y, \omega_x, \omega_y\}$;
(2) Fully learnable $\mathbf{P}^C$ and analytic $\mathbf{\Phi}_{xy}^{\text{batch}}$;
(3) Fully learnable $\mathbf{P}^C$ and $\mathbf{E}$ but keeping the structure imposed by $\mathbf{C}$ and $\mathbf{S}$, with learnable rotational frequencies $\boldsymbol{\omega}_x, \boldsymbol{\omega}_y$. We impose constraints $\mathbf{P}^C > \mathbf{0}$, to ensure positive definite precisions, and $\mathbf{E} \in (\mathbf{0}, \mathbf{1}]$ to ensure the system remains stable.

IV. Model Architecture

A. Hybrid Encoder-Decoder Architecture

To capture features across spatial scales, the encoder follows a U-Net design, but all standard convolutions are replaced with AFA layers, so local feature processing encodes stability and uncertainty fusion. Each frame is processed individually, and the resulting patch features are aggregated, flattened, and treated as tokens for downstream Transformers, which are placed at every U-Net level. This balances the local inductive bias of convolution with the global spatio-temporal reasoning of Transformers. The decoder upsamples with standard transpose convolutions.

B. The Sliding Window Attention (SWA) Block

The SWA Block is a regularization method for short sequences. It exposes a fixed non-causal self-attention module to every causal subsequence by shifting and zero-padding the temporal dimension (e.g., $[\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3] \to [\mathbf{0}, \mathbf{I}_1, \mathbf{I}_2]$). This T-fold expansion is not scalable for long horizons but is tractable for small T and enforces a temporal stationarity prior: attention weights must remain consistent across shifts, so the model learns time-invariant dynamics.

This scheme both enforces causality and provides implicit data augmentation, since each shifted subsequence defines a prediction task. Because causality is imposed externally, the attention block itself can remain fully non-causal, letting tokens attend to all spatial and temporal neighbors within a window. In effect, SWA decouples causal structure from attention, improving generalization and stability.

C. State Estimation and Prediction

A key principle in optimal filtering is separating state estimation (filtering) from prediction (forecasting). The encoder produces a current-state estimate $\mathbf{x}_{t|t}$ by fusing the full input history with non-causal self-attention. An auxiliary reconstruction loss on the current frame ensures the estimate retains high-fidelity spatial detail.

Prediction is handled by the Dynamics Prediction Module, a small U-Net with non-causal attention at each level. This module takes the causally constrained encoder features and produces the one-step-ahead forecast ($\mathbf{x}_{t+1|t}$). The total training loss includes both a reconstruction term on the encoder's state estimate and a one-step prediction term, ensuring that the representation preserves information about the present frame while also accurately forecasting the next.

D. Implementation

Discuss size number of layers in the U-Net; of the model, number of parameters, etc.

The dataset consists of X images, with masks of GA regions. We append the original image, the mask, and the mask residual (the growth region, computed by subtracting the previous mask from the current one). Each is of size 256×256, so altogether the data is of size 256×768. We give this whole image as the input to the network. We pretrain the U-Net autoencoder, then train the full spatio-temporal architecture. We use a hybrid loss of reconstruction and prediction, or we can just predict the growth region. Or we include the reconstruction loss in a pretraining phase, and then finetune on just prediction of the growth region. We use a Dice Loss, on the border region only.

Data augmentation includes random shifts, rotations, flips, color jitter, and additive zero-mean Gaussian white noise. We use the Adam optimizer with a learning rate of 1E-4. The model is evaluated using X-fold cross-validation.

V. Experiments

A. Baseline Models and Ablations

To isolate the contribution of the SDE structure ($\mathbf{\Phi}$) and the optimal fusion mechanism ($\mathbf{P}^C$), we establish two primary performance baselines and two key architectural ablations. All models utilize the identical U-Net and DynNet architecture, differing only in the design and parameterization of the local convolutional layer.

1) Projected Depthwise Convolution (PDWConv): This model serves as the minimal, unstructured baseline. A standard 3×3 Depthwise Convolution (DWConv) is utilized, wrapped by 1×1 projection layers to match the input/output dimensionality of the AFA kernel. This model is non-adaptive and unstructured, measuring the minimal performance achievable by a local feature extractor.

2) Simple Local Self-Attention (Simple LSA): This baseline measures the generic benefit of adaptivity. A standard 3×3 local Self-Attention block is implemented using the Euclidean dot-product score ($\mathbf{Q} \cdot \mathbf{K}^T$) and a Softmax normalization denominator. This removes the AFA's probabilistic structure and optimal fusion mechanism, providing a direct comparison against a plain, adaptive similarity scoring model.

3) AFA Ablation: Real-Valued Dynamics: This ablation tests the unique value of the complex rotational structure ($\mathbb{C}^d$) derived from the SDE. The complex projections and the transition matrix $\mathbf{\Phi}$ are constrained to be purely real ($\mathbf{\Phi}_i = \mathbf{0}$). This removes the ability to model directional flow and complex coupling, measuring whether the superior generalization of the AFA kernel relies specifically on the expressive power of rotation or if simple real-valued exponential decay is sufficient.

4) AFA Ablation: Standard RoPE Positional Encoding: This ablation isolates the structural benefit of the SDE's analytical derivation. The dynamic calculation of $\mathbf{\Phi}$ is replaced entirely by the application of a fixed, standard 2D Rotary Positional Embedding (RoPE) before the local dot-product score. This tests whether the SDE structure provides a superior, generalized functional prior for dynamics compared to a widely adopted, fixed positional bias mechanism.

B. Results on CIFAR-10

1. Standard classification (e.g., MNIST or CIFAR-10) using a minimal backbone (e.g., a simple ResNet-style block or a shallow CNN).

2. Tabulate accuracy and loss for the three variants of AFA and three baseline core models (all fully learned): (AFA Conv (analytic, partially analytic, fully learned), PDWConv, Simple LSA, AFA Real-Valued). Proves that the $\mathbf{\Phi}$ dynamics (complex rotation) and the $\mathbf{P}^C$ fusion (optimal estimate) significantly outperform unstructured local kernels and simple Softmax attention.

3. Report the measured latency of the AFA layer versus PDWConv (e.g., the $3.5\times$ to $5\times$ factor). (Reasonable params are kernel size of 3 to 5 and embedding dimension of d=32).

VI. Discussion and Conclusion

References

[1] Peter Racioppo. Attention as an adaptive filter, 2025.

[2] Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between gaussian fields and gaussian markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

[3] Per Sidén and Fredrik Lindsten. Deep gaussian markov random fields, 2020.

[4] Taco S. Cohen and Max Welling. Group equivariant convolutional networks, 2016.

[5] Maurice Weiler and Gabriele Cesa. General $e(2)$-equivariant steerable cnns, 2021.

[6] Andrey Alekseev and Anatoly Bobe. Gabornet: Gabor filters with learnable parameters in deep convolutional neural networks, 2019.

[7] Suhyuk Um, Jaeyoon Kim, and Dongbo Min. Fast 2-d complex gabor filter with kernel decomposition, 2017.

[8] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition, 2019.

[9] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks, 2020.

[10] Dennis Gramlich, Patricia Pauli, Carsten W. Scherer, Frank Allgöwer, and Christian Ebenbauer. Convolutional neural networks as 2-d systems, 2023.

[11] Rahul G. Krishnan, Uri Shalit, and David Sontag. Deep kalman filters, 2015.

[12] Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. Kalmannet: Neural network aided kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022.

[13] Jiaming Wang, Xinyu Geng, and Jun Xu. Nonlinear kalman filtering based on self-attention mechanism and lattice trajectory piecewise linear approximation, 2024.

[14] Siyuan Shen, Jichen Chen, Guanfeng Yu, Zhengjun Zhai, and Pujie Han. Kalmanformer: using Transformer to model the Kalman gain in Kalman filters. *Frontiers in Neurorobotics*, 18:1460255, 2025.

[15] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.

[16] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, September 2024.

[17] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer, 2024.

[18] Avinash Amballa. Cope: A lightweight complex positional encoding, 2025.

[19] Jintian Shao, Hongyi Huang, Jiayi Wu, Beiwen Zhang, ZhiYu Wu, You Shan, and MingKai Zheng. ComplexFormer: Disruptively advancing Transformer inference ability via head-specific complex vector attention, 2025.

[20] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022.

[21] 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, 2021.

[22] Eric Nguyen, Karan Goel, Albert Gu, Gordon W. Downs, Preey Shah, Tri Dao, Stephen A. Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals using state spaces, 2022.

[23] Ethan Baron, Itamar Zimerman, and Lior Wolf. 2-d ssm: A general spatial layer for visual transformers, 2023.

[24] Patricia Pauli, Dennis Gramlich, and Frank Allgöwer. State space representations of the roesser type for convolutional layers, 2024.

[25] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks, 2016.

[26] Brandon Yang, Gabriel Bender, Quoc V. Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference, 2020.

[27] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models, 2019.

[28] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer, 2023.

[29] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers, 2020.

[30] Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition, 2021.

[31] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. On the connection between local attention and dynamic depth-wise convolution, 2022.

[32] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers, 2021.

[33] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network, 2022.

[34] Kin Wai Lau, Lai-Man Po, and Yasar Abbas Ur Rehman. Large separable kernel attention: Rethinking the large kernel attention design in cnn, 2023.

[35] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc V. Le. Transformer quality in linear time, 2022.

[36] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017.

[37] Stefan K. Nielsen, Laziz U. Abdullaev, Rachel S. Y. Teo, and Tan M. Nguyen. Elliptical attention, 2024.