PFGNet: A Fully Convolutional Frequency-Guided Peripheral Gating Network for Efficient Spatiotemporal Predictive Learning

Supplementary Material

1. Existence and Optimality of Ring-Shaped Pass Band

In the main text, we established a **strong existence** result under monotonicity, guaranteeing a ring-shaped pass band for $\beta_k \in (0, \beta_{\max})$. Here we present a **weak existence** theorem that aligns with PFGNet's implementation, where $\beta = \tanh(\beta_{\text{raw}}) \in (-1, 1)$.

Theorem 1 (Weak Existence of Ring-Shaped Pass Band)**.** *Let $H_1, H_2 : [0, \pi] \to \mathbb{R}$ denote continuous radial frequency-response profiles. Define $f(r) = H_1(r) - \beta H_2(r)$ for $\beta \in (-1, 1)$. Assume there exist $0 \le c < a < b \le \pi$ such that*

$$f(c) \le 0, \quad f(a) > 0, \quad f(b) \le 0.$$

Then there exist r_1 and r_2 with $c \le r_1 < a < r_2 \le b$ such that

$$f(r) > 0 \quad \text{for all } r \in (r_1, r_2), \text{ and } f(r_1) = f(r_2) = 0.$$

Thus $H_\beta(\omega) = H_1(\|\omega\|) - \beta H_2(\|\omega\|)$ has a non-degenerate ring-shaped pass band $\{\omega : r_1 < \|\omega\| < r_2\}$.

Proof. Since f is continuous with $f(c) \le 0$ and $f(a) > 0$, the set $S_1 = \{r \in [c, a] : f(r) \le 0\}$ is non-empty and closed. We define $r_1 = \max S_1$. Clearly, $c \le r_1 < a$ and $f(r_1) = 0$. By the definition of the maximum, $f(r) > 0$ for all $r \in (r_1, a]$.

Similarly, the set $S_2 = \{r \in [a, b] : f(r) \le 0\}$ is non-empty and closed because $f(a) > 0$ and $f(b) \le 0$. We define $r_2 = \min S_2$. Thus $a < r_2 \le b$ and $f(r_2) = 0$. By the definition of the minimum, $f(r) > 0$ for all $r \in [a, r_2)$.

Combining these results, we have $f(r) > 0$ on the continuous interval (r_1, r_2). □

In PFGNet, we set:

- H_1: frequency response of a large-kernel convolution,
- H_2: frequency response of a small-kernel convolution,
- $\beta = \tanh(\beta_{\text{raw}}) \in (-1, 1)$: bounded center-suppression coefficient.

The large kernel decays slowly from DC to mid-frequencies, while the small kernel decays rapidly. Their difference $f(r) = H_1(r) - \beta H_2(r)$ naturally satisfies:

- $f(c) \le 0$ for small $c > 0$ (small kernel dominates near DC),
- $f(a) > 0$ in mid-frequencies (large kernel retains more energy),
- $f(b) \le 0$ near $b \approx \pi$ (both decay to zero).

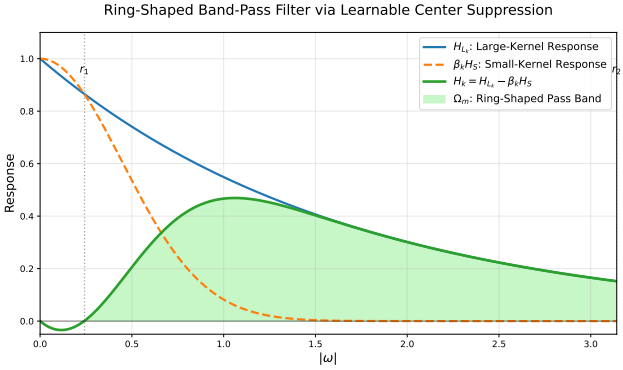


Figure 1. Frequency response of the PFG block. The large-kernel response H_{L_k} decays slowly, while the small-kernel response H_S decays rapidly. Their difference $H_k = H_{L_k} - \beta_k H_S$ (with β_k=0.75) is positive within the green region, forming a ring-shaped band-pass filter as stated in the main text.

Hence, the combination of large- and small-kernel convolutions ensures the sign pattern required by the corrected weak theorem, yielding a non-degenerate ring-shaped pass band in practice. We illustrate the frequency-selective behavior of the PFG block in Figure 1. Using a large kernel with a slowly decaying frequency response H_{L_k} (e.g., exponential decay with rate 0.6) and a small kernel with a rapidly decaying response H_S (e.g., Gaussian decay with variance parameter 2.5), their difference $H_k = H_{L_k} - \beta_k H_S$ (with $\beta_k = 0.75$) suppresses low-frequency background near DC, amplifies mid-frequency motion cues, and attenuates high-frequency noise, forming a ring-shaped band-pass filter as predicted by our theoretical analysis.

Furthermore, we have the following theorem:

Theorem 2 (Existence of an SNR-maximizing $\beta^\star$)**.** *Assume the input signal has spectral power $P_S(\omega) \ge 0$ and the additive noise is white with constant power $P_N(\omega) \equiv \sigma_N^2 > 0$. Let the composite filter response be $H_\beta = H_L - \beta H_S$, where H_L and H_S denote the frequency responses of the large and small kernels, respectively. Define the signal-to-noise ratio*

$$\mathrm{SNR}(\beta) = \frac{\int |H_\beta(\omega)|^2 P_S(\omega)\, d\omega}{\int |H_\beta(\omega)|^2 P_N(\omega)\, d\omega}.$$

If H_L and H_S are linearly independent in $L^2([0, \pi])$ and $\int |H_S|^2 P_S\, d\omega > 0$, then $\mathrm{SNR}(\beta)$ admits at least one finite stationary point $\beta^\star$ satisfying $\frac{d}{d\beta}\mathrm{SNR}(\beta) = 0$.

Proof. Define the signal energy $N(\beta)$ and noise energy

$D(\beta)$ as quadratic forms:

$$N(\beta) := \int |H_L - \beta H_S|^2 P_S(\omega)\, d\omega = A - 2\beta B + \beta^2 C,$$

$$D(\beta) := \int |H_L - \beta H_S|^2 P_N(\omega)\, d\omega = \sigma_N^2 \left(\tilde{A} - 2\beta\tilde{B} + \beta^2\tilde{C}\right),$$

where the coefficients are given by the respective integrals (e.g., $C = \int |H_S|^2 P_S\, d\omega$ and $\tilde{C} = \int |H_S|^2\, d\omega$).

By the linear independence of H_L and H_S in L^2, the denominator $D(\beta)$ is strictly positive for all $\beta \in \mathbb{R}$, ensuring that $\mathrm{SNR}(\beta) = \frac{N(\beta)}{D(\beta)}$ is continuously differentiable on the entire real line.

Consider the asymptotic behavior of $\mathrm{SNR}(\beta)$ as $\beta \to \pm\infty$. Dividing the numerator and denominator by β^2 yields:

$$\lim_{\beta\to\pm\infty} \mathrm{SNR}(\beta) = \lim_{\beta\to\pm\infty} \frac{A/\beta^2 - 2B/\beta + C}{\sigma_N^2(\tilde{A}/\beta^2 - 2\tilde{B}/\beta + \tilde{C})} = \frac{C}{\sigma_N^2\tilde{C}}.$$

By assumption, $C > 0$ and $\tilde{C} > 0$, so this limit is a finite positive constant, which we denote as L.

We now examine the continuous function $\mathrm{SNR}(\beta)$. There are three possible cases: 1. $\mathrm{SNR}(\beta) \equiv L$ for all β. Then every $\beta \in \mathbb{R}$ is a stationary point, and the theorem holds trivially. 2. There exists some $\beta_0 \in \mathbb{R}$ such that $\mathrm{SNR}(\beta_0) > L$. Since the function is continuous and decays to L at both infinities, by the Extreme Value Theorem, $\mathrm{SNR}(\beta)$ must attain a global maximum at some finite point $\beta^\star$. 3. There exists some $\beta_0 \in \mathbb{R}$ such that $\mathrm{SNR}(\beta_0) < L$. By the same reasoning, $\mathrm{SNR}(\beta)$ attains a global minimum at some finite point $\beta^\star$.

In cases 2 and 3, Fermat's theorem guarantees that at the local extremum $\beta^\star$, the derivative must vanish: $\frac{d}{d\beta}\mathrm{SNR}(\beta^\star) = 0$.

Furthermore, setting the derivative to zero yields:

$$N'(\beta)D(\beta) - N(\beta)D'(\beta) = 0.$$

Expanding this expression, the β^3 terms mathematically cancel out ($C\tilde{C}\beta^3 - \tilde{C}C\beta^3 = 0$), leaving a polynomial equation of degree at most 2:

$$(B\tilde{C} - C\tilde{B})\beta^2 + (C\tilde{A} - A\tilde{C})\beta + (\tilde{B}A - B\tilde{A}) = 0.$$

This confirms that $\mathrm{SNR}(\beta)$ has at most two finite stationary points, and we have proven that at least one must exist. □

Lemma 1 (SNR Advantage of $H_L - \beta H_S$ over H_L)**.** *Assume the input signal has spectral power $P_S(\omega) \geq 0$ and additive white noise with power $\sigma_N^2 > 0$. Let H_L and H_S be the frequency responses of the large and small kernels, respectively, and assume they are linearly independent in $L^2([0,\pi])$. Define the composite filter $H_\beta = H_L - \beta H_S$ and the SNR as*

$$\mathrm{SNR}(\beta) = \frac{\int |H_\beta|^2 P_S\, d\omega}{\int |H_\beta|^2 \sigma_N^2\, d\omega}.$$

Assume further that H_L is not already a stationary point of the SNR in the direction of H_S, meaning

$$\int \mathrm{Re}(H_L\overline{H_S})P_S\, d\omega \int |H_L|^2\, d\omega \neq \int |H_L|^2 P_S\, d\omega \int \mathrm{Re}(H_L\overline{H_S})\, d\omega$$

(i.e., $B\tilde{A} \neq A\tilde{B}$ in the notation below). Then, there exists $\hat{\beta} \neq 0$ such that

$$\mathrm{SNR}(\hat{\beta}) > \mathrm{SNR}(0).$$

Proof. Define the energy integrals

$$N(\beta) = A - 2\beta B + \beta^2 C, \qquad D(\beta) = \sigma_N^2(\tilde{A} - 2\beta\tilde{B} + \beta^2\tilde{C}),$$

so that $\mathrm{SNR}(\beta) = N(\beta)/D(\beta)$ and $\mathrm{SNR}(0) = A/(\sigma_N^2\tilde{A})$. Consider the difference numerator:

$$\Delta(\beta) := N(\beta)\tilde{A} - A\frac{D(\beta)}{\sigma_N^2} = -2\beta(B\tilde{A} - A\tilde{B}) + \beta^2(C\tilde{A} - A\tilde{C}).$$

Clearly $\Delta(0) = 0$. By our assumption, the coefficient of the linear term $K = -2(B\tilde{A} - A\tilde{B})$ is strictly non-zero. For values of β sufficiently close to 0, the linear term $K\beta$ strictly dominates the quadratic term $\beta^2(C\tilde{A} - A\tilde{C})$. Since $K \neq 0$, we can choose a $\hat{\beta}$ with a sufficiently small absolute value and the same sign as K. This guarantees that $\Delta(\hat{\beta}) \approx K\hat{\beta} > 0$.

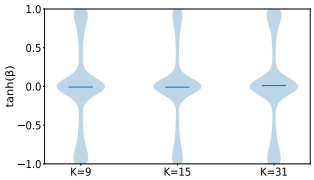
Since $D(\hat{\beta}) > 0$ and $\tilde{A} > 0$, the inequality $\Delta(\hat{\beta}) > 0$ directly implies

$$N(\hat{\beta})\tilde{A} > A\frac{D(\hat{\beta})}{\sigma_N^2} \implies \frac{N(\hat{\beta})}{D(\hat{\beta})} > \frac{A}{\sigma_N^2\tilde{A}}.$$

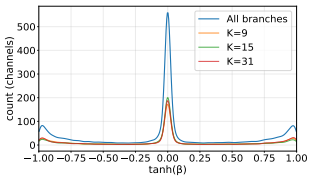
Thus, $\mathrm{SNR}(\hat{\beta}) > \mathrm{SNR}(0)$. □

In PFGNet, we learn a per-channel β in each branch and share it across spatial locations via backpropagation. Since local spectral content varies across pixels, such as motion versus static regions, pixel-wise softmax fusion enables spatially varying optimal denoising, while the channel-wise β provides learnable center suppression, outperforming any fixed global suppression. Moreover, subtracting βH_S can be beneficial: by Lemma 1, there exists a coefficient $\hat{\beta} \neq 0$ such that the filter $H_L - \hat{\beta}H_S$ achieves a strictly higher SNR than the plain large-kernel filter H_L.

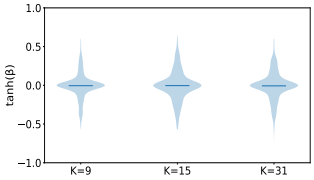
This is empirically corroborated in Figure 2. On Moving MNIST, learned channel-wise $\tanh(\beta)$ values are symmetrically distributed around zero across all branches (K=9, 15, 31), with significant mass at ± 1, indicating that both enhancement and suppression of peripheral responses are utilized to model simple digit motion. On TaxiBJ, a complex traffic dataset, $\tanh(\beta)$ is centered at zero with reduced variance for larger kernels, reflecting balanced and mild center modulation to preserve rich mid-frequency flow patterns. Notably, the symmetric usage of positive and negative β on Moving MNIST and the near-zero mean on



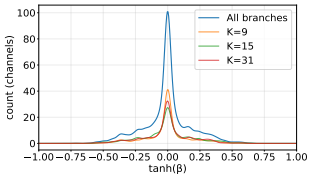
(a) tanh(β) by branch distribution on Moving MNIST



(b) channel-wise tanh(β) value distribution on Moving MNIST



(c) tanh(β) by branch distribution on TaxiBJ



(d) channel-wise tanh(β) value distribution on TaxiBJ

Figure 2. Visualization of $\tanh(\beta)$ on Moving MNIST and TaxiBJ datasets. Each pair shows (a,c) the branch-wise $\tanh(\beta)$ distributions and (b,d) the smoothed channel-wise count curves, where the horizontal axis denotes $\tanh(\beta)$ values and the vertical axis indicates the number of channels.

TaxiBJ are consistent with our SNR analysis: the learned channel-wise β adapts in sign and magnitude to local spectral statistics, supporting the view that pixel-wise softmax gating together with channel-wise suppression improves denoising flexibility in practice.

2. Metric Definitions

Notation Let the prediction and ground truth be $\hat{\mathbf{Y}}, \mathbf{Y} \in \mathbb{R}^{N\times T\times C\times H\times W}$. Dimensions: N is the batch size, T is the temporal length, C is the number of channels, and H, W are the spatial sizes. An element $\hat{\mathbf{Y}}_{n,t,c,h,w}$ (or $\mathbf{Y}_{n,t,c,h,w}$) is the value at batch index n, time step t, channel c, and spatial location (h, w). Define the per-frame spatial size (including channels) as $S = CHW$. All scalar metrics are averaged over the batch and time dimensions (N, T).

2.1. Mean Squared Error (MSE)

Non-spatially normalized

$$\mathrm{MSE} = \sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(\frac{1}{NT}\sum_{n=1}^{N}\sum_{t=1}^{T}(\hat{\mathbf{Y}}_{n,t,c,h,w} - \mathbf{Y}_{n,t,c,h,w})^2\right).$$

Spatially normalized

$$\mathrm{MSE}_{\mathrm{norm}} = \frac{1}{NT}\sum_{n=1}^{N}\sum_{t=1}^{T}\left(\frac{1}{S}\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}(\hat{\mathbf{Y}}_{n,t,c,h,w} - \mathbf{Y}_{n,t,c,h,w})^2\right).$$

In practice, the spatially normalized MSE is used during training and validation, while the non-spatially normalized version is reported during testing.

2.2. Mean Absolute Error (MAE)

$$\mathrm{MAE} = \sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(\frac{1}{NT}\sum_{n=1}^{N}\sum_{t=1}^{T}|\hat{\mathbf{Y}}_{n,t,c,h,w} - \mathbf{Y}_{n,t,c,h,w}|\right).$$

2.3. Peak Signal-to-Noise Ratio (PSNR)

Each pixel value a is converted to 8-bit as $\tilde{a} = \mathrm{uint8}(255\,a)$. For each sample frame (n, t), the per-pixel mean squared error is

$$\mathrm{MSE}_{n,t} = \frac{1}{S}\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}(\tilde{\hat{\mathbf{Y}}}_{n,t,c,h,w} - \tilde{\mathbf{Y}}_{n,t,c,h,w})^2.$$

With the dynamic range upper bound $I_{\max} = 255$, the frame-level and final PSNR are

$$\mathrm{PSNR}_{n,t} = 20\log_{10}(I_{\max}) - 10\log_{10}(\mathrm{MSE}_{n,t}),$$

$$\mathrm{PSNR} = \frac{1}{NT}\sum_{n=1}^{N}\sum_{t=1}^{T}\mathrm{PSNR}_{n,t}.$$

2.4. Structural Similarity (SSIM)

For each frame, skimage.metrics.structural_similarity is applied in the $[0, 1]$ floating domain and averaged over (N, T). The frame-level SSIM uses the standard form. With a Gaussian window G,

$$\mu_x = G * x, \quad \mu_y = G * y, \quad \sigma_x^2 = G * (x^2) - \mu_x^2,$$
$$\sigma_y^2 = G * (y^2) - \mu_y^2, \quad \sigma_{xy} = G * (xy) - \mu_x\mu_y,$$

where x and y denote two corresponding image patches, and $\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \sigma_{xy}$ are their local means, variances, and covariance computed using the Gaussian window. The SSIM for a pair of patches is defined as

$$\mathrm{SSIM}_{\mathrm{patch}}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)},$$

where the dynamic range upper bound is $I_{\max} = 1$ and $C_1 = (0.01\,I_{\max})^2$, $C_2 = (0.03\,I_{\max})^2$. The SSIM of an entire frame $\hat{\mathbf{Y}}_{n,t}, \mathbf{Y}_{n,t}$ is obtained by averaging over all overlapping patches:

$$\mathrm{SSIM}_{\mathrm{frame}}\left(\hat{\mathbf{Y}}_{n,t}, \mathbf{Y}_{n,t}\right) = \frac{1}{|\mathcal{P}|}\sum_{(x,y)\in\mathcal{P}}\mathrm{SSIM}_{\mathrm{patch}}(x, y),$$

where $\mathcal{P}$ denotes the set of all patches. The final SSIM is

$$\mathrm{SSIM} = \frac{1}{NT}\sum_{n=1}^{N}\sum_{t=1}^{T}\mathrm{SSIM}_{\mathrm{frame}}\left(\hat{\mathbf{Y}}_{n,t}, \mathbf{Y}_{n,t}\right).$$

3. Complete experimental results

In the main text, we presented representative results for clarity. Here, we report complete quantitative comparisons in

Table 1. Quantitative comparison on the Moving MNIST dataset.

Method	MSE ↓	SSIM ↑
Recurrent-based Methods		
ConvLSTM [21]	103.3	0.707
DFN [7]	89.0	0.726
FRNN [19]	69.7	0.813
VPN [10]	64.1	0.870
PredRNN [29]	56.8	0.867
CausalLSTM [30]	46.5	0.898
MIM [32]	44.2	0.910
E3D-LSTM [31]	41.3	0.910
LMC [12]	41.5	0.924
MAU [1]	27.6	0.937
PhyDNet [4]	24.4	0.947
CrevNet [34]	22.3	0.949
SwinLSTM [25]	17.7	0.962
VMRNN [26]	16.5	0.965
Recurrent-free Methods		
SimVP [3]	23.8	0.948
MMVP [36]	22.2	0.952
TAU [22]	19.8	0.957
PFGNet(Ours)	**15.2**	**0.967**

Table 2. Quantitative comparison on the TaxiBJ dataset.

Method	Params	FLOPs	MSE ↓	MAE ↓	SSIM ↑
Recurrent-based Methods					
ConvLSTM [21]	15.0M	20.7G	0.3358	15.32	0.9836
PredNet [18]	12.5M	0.9G	0.3516	15.91	0.9828
PredRNN [29]	23.7M	42.4G	0.3194	15.31	0.9838
PredRNN++ [30]	38.4M	63.0G	0.3348	15.37	0.9834
E3D-LSTM [31]	51.0M	98.2G	0.3421	14.98	0.9842
PhyDNet [4]	3.1M	5.6G	0.3622	15.53	0.9828
MIM [32]	37.9M	64.1G	0.3110	14.96	0.9847
MAU [1]	4.4M	6.4G	0.3062	15.26	0.9840
PredRNNv2 [33]	23.7M	42.6G	0.3834	15.55	0.9826
SwinLSTM [25]	2.9M	1.3G	0.3026	15.00	0.9843
VMRNN [26]	2.6M	0.9G	0.2887	**14.69**	**0.9858**
Recurrent-free Methods					
SimVP [3]	13.8M	3.6G	0.3282	15.45	0.9835
TAU [22]	9.6M	2.5G	0.3108	14.93	0.9848
SimVPv2 [24]	10.0M	2.6G	0.3246	15.03	0.9844
Met2Net [15]	16.8M	8.8G	0.3164	14.82	0.9851
PFGNet(Ours)	**1.9M**	**0.6G**	**0.2881**	14.75	0.9857

Table 3. Quantitative comparison on the KTH dataset.

Method	KTH (10 → 20)		KTH (10 → 40)	
	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑
Recurrent-based Methods				
ConvLSTM [21]	0.712	23.58	0.639	22.85
SAVP [11]	0.746	25.38	0.701	23.97
FRNN [19]	0.771	26.12	0.678	23.77
DFN [7]	0.794	27.26	0.652	23.01
PredRNN [29]	0.839	27.55	0.703	24.16
VarNet [8]	0.843	28.48	0.739	25.37
SAVP-VAE [11]	0.852	27.77	0.811	26.18
PredRNN++ [30]	0.865	28.47	0.741	25.21
E3D-LSTM [31]	0.879	29.31	0.810	27.24
STMFANet [9]	0.893	29.85	0.851	27.56
SwinLSTM [25]	0.903	**34.34**	0.879	**33.15**
VMRNN [26]	0.907	34.06	0.882	32.69
Recurrent-free Methods				
SimVP [3]	0.905	33.72	0.886	32.93
MMVP [36]	0.906	27.54	0.888	26.35
PFGNet(Ours)	**0.911**	34.10	**0.891**	32.64

Table 4. Quantitative comparison on the Human3.6M dataset.

Method	Params	FLOPs	MSE ↓	MAE ↓	SSIM ↑
Recurrent-based Methods					
ConvLSTM [21]	15.5M	347.0G	125.5	1566.7	0.9813
E3D-LSTM [31]	60.9M	542.0G	143.3	1442.5	0.9803
PredNet [18]	12.5M	**13.7G**	261.9	1625.3	0.9786
PhyDNet [4]	4.2M	19.1G	125.7	1614.7	0.9804
MAU [1]	20.2M	105.0G	127.3	1577.0	0.9812
MIM [32]	47.6M	1051.0G	112.1	1467.1	0.9829
PredRNN [29]	24.6M	704.0G	113.2	1458.3	0.9831
PredRNN++ [30]	39.3M	1033.0G	110.0	1452.2	0.9832
PredRNNv2 [33]	24.6M	708.0G	114.9	1484.7	0.9827
Recurrent-free Methods					
SimVP [3]	41.2M	197.0G	115.8	1511.5	0.9822
SimVPv2 [24]	11.3M	74.6G	**108.4**	1441.0	0.9834
TAU [22]	37.6M	182.0G	113.3	**1390.7**	**0.9839**
ViT [2]	28.3M	239.0G	136.3	1603.5	0.9796
Swin Transformer [16]	38.8M	188.0G	133.2	1599.7	0.9799
Uniformer [13]	27.7M	211.0G	116.3	1497.7	0.9824
MLP-Mixer [27]	47.0M	164.0G	125.7	1511.9	0.9819
ConvMixer [28]	**3.1M**	39.4G	115.8	1527.4	0.9822
Poolformer [35]	31.2M	156.0G	118.4	1484.1	0.9827
ConvNeXt [17]	31.4M	157.0G	113.4	1469.7	0.9828
VAN [5]	37.5M	182.0G	111.4	1454.5	0.9831
HorNet [20]	28.1M	143.0G	118.1	1481.1	0.9824
MogaNet [14]	8.6M	63.6G	109.1	1446.4	0.9834
PFGNet(Ours)	7.3M	58.3G	111.3	1392.4	0.9838

Tables 1, 2, 3, and 4, covering all baseline methods from prior literature. Models that exhibit substantially lower performance than PFGNet were omitted from the main text for readability but are included here for completeness.

Furthermore, we extend our evaluation to the Moving

Table 5. Quantitative comparison on the Moving FMNIST dataset.

Method	Params	FLOPs	MSE ↓	MAE ↓	SSIM ↑
Recurrent-based Methods					
ConvLSTM-S [21]	15.0M	56.8G	28.87	113.20	0.8793
ConvLSTM-L [21]	33.8M	127.0G	25.51	104.85	0.8928
PredNet [18]	12.5M	8.6G	185.94	318.30	0.6713
PhyDNet [4]	**3.1M**	15.3G	34.75	125.66	0.8567
PredRNN [29]	23.8M	116.0G	22.01	**91.74**	0.9091
PredRNN++ [30]	38.6M	171.7G	**21.71**	91.97	**0.9097**
MIM [32]	38.0M	179.2G	23.09	96.37	0.9043
MAU [1]	4.5M	17.8G	26.56	104.39	0.8722
E3D-LSTM [31]	51.0M	298.9G	35.35	110.09	0.8722
PredRNNv2 [33]	23.9M	116.6G	24.13	97.46	0.9004
DMVFN [6]	3.5M	**0.2G**	118.32	220.02	0.7572
Recurrent-free Methods					
SimVP [3]	58.0M	19.4G	30.77	113.94	0.8740
SimVPv2 [24]	46.8M	16.5G	25.86	101.22	0.8933
TAU [22]	44.7M	16.0G	24.24	96.72	0.8995
ViT [2]	46.1M	16.9G	31.05	115.59	0.8712
Swin Transformer [16]	46.1M	16.4G	28.66	108.93	0.8815
Uniformer [13]	44.8M	16.5G	29.56	111.72	0.8779
MLP-Mixer [27]	38.2M	14.7G	28.83	109.51	0.8803
ConvMixer [28]	3.9M	5.5G	31.21	115.74	0.8709
Poolformer [35]	37.1M	14.1G	30.02	103.39	0.8750
ConvNeXt [17]	37.3M	14.1G	26.41	102.56	0.8908
VAN [5]	44.5M	16.0G	31.39	116.28	0.8823
HorNet [20]	45.7M	16.3G	29.19	110.17	0.8798
MogaNet [14]	46.8M	16.5G	25.14	99.69	0.8963
PFGNet(Ours)	41.3M	15.2G	23.55	94.78	0.9024

Fashion-MNIST (MFMNIST) dataset, with baseline results obtained from OpenSTL [23] under identical experimental settings. Figure 3 presents qualitative results of PFGNet. As shown in Table 5, PFGNet achieves the *highest accuracy among all recurrent-free models* (MSE: 23.55, SSIM: 0.9024) while maintaining moderate complexity (41.3M parameters, 15.2G FLOPs). Notably, compared to recurrent-based PredRNN (116.0G FLOPs) and PredRNN++ (171.7G FLOPs), PFGNet delivers *significantly lower computational cost* with comparable predictive performance, highlighting its efficiency in spatiotemporal forecasting.

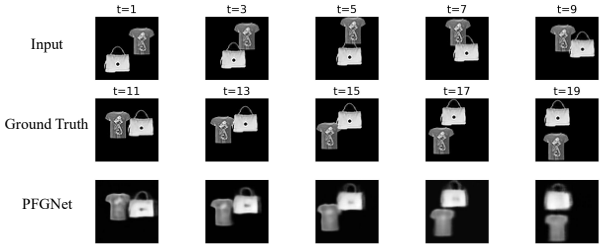


Figure 3. Qualitative results of PFGNet on Moving FMNIST.

4. Additional Ablation Studies

As shown in Figure 4, we further analyze the effect of different n settings in the asymmetric convolution ($n \times k + k \times n$) on the TaxiBJ dataset. The results show that varying n brings only minor changes in performance; setting n=1

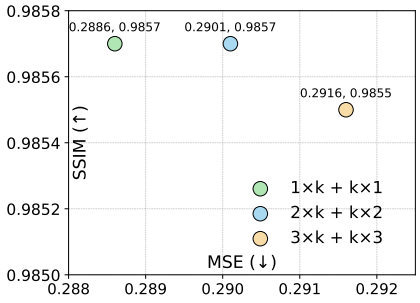


Figure 4. Ablation on asymmetric convolution ($n \times k + k \times n$).

Table 6. Ablation on the number of PFG blocks N_t on Moving MNIST and TaxiBJ.

Dataset	N_t	Params	FLOPs	MSE ↓	MAE ↓	SSIM ↑
Moving MNIST (100 epochs)	2	11.0M	7.4G	34.70	93.93	0.9201
	4	21.1M	10.0G	29.74	83.38	0.9326
	6	31.2M	12.6G	28.33	80.14	0.9363
	8	41.3M	15.2G	27.61	78.54	0.9381
	10	51.3M	17.7G	**27.21**	**77.57**	**0.9391**
TaxiBJ	2	0.53M	0.21G	0.3669	15.88	0.9826
	4	0.97M	0.33G	0.3216	15.21	0.9842
	6	1.41M	0.44G	0.3130	14.99	0.9851
	8	1.86M	0.55G	**0.2881**	**14.75**	**0.9857**
	10	2.30M	0.67G	0.2919	14.78	0.9853

Table 7. Ablation on the number of PFG blocks N_t on KTH (10→20).

Dataset	N_t	Params	FLOPs	SSIM ↑	PSNR ↑
KTH 10 → 20	2	10.8M	57.1G	0.908	33.88
	4	20.8M	98.5G	0.910	34.01
	6	30.9M	0.14T	0.911	34.10
	8	41.0M	0.18T	0.909	34.10
	10	51.1M	0.22T	**0.912**	**34.24**

achieves nearly the same accuracy while reducing computational complexity. Therefore, we adopt n=1 as a compact and efficient default configuration in our experiments.

In all main experiments, we set the number of PFG blocks N_t to follow the same configuration as SimVP and its variants in OpenSTL to ensure a fair comparison; Tables 6 and 7 further investigate how varying N_t from 2 to 10 affects model complexity and accuracy. As expected, Params and FLOPs grow linearly with N_t, since stacking more PFG blocks only introduces additional, structurally identical modules. On Moving MNIST, a larger N_t consistently leads to lower MSE/MAE and higher SSIM, indicating that this dataset benefits from deeper temporal modeling and does not show overfitting within the tested range. On TaxiBJ, performance also improves as N_t increases, but the best results are obtained at N_t=8, while N_t=10 brings slightly worse scores, suggesting a mild overfitting

tendency and diminishing returns when the model becomes too deep. On KTH (10→20), increasing N_t from 2 to 10 generally leads to higher SSIM and PSNR, though the improvements become marginal once $N_t \geq 6$ and even show slight fluctuations, and the final training loss remains clearly below the validation loss, indicating that the motion structure is already well captured and extra depth mainly refines local details rather than improving generalization. In practice, choosing N_t=4 or N_t=6 recovers most of the gains while keeping the model compact, which aligns with the settings commonly used in OpenSTL. Overall, these results confirm that the OpenSTL default choice is well justified and provides a well-balanced trade-off between computational efficiency and predictive performance, and we therefore adopt this setting in our main experiments.

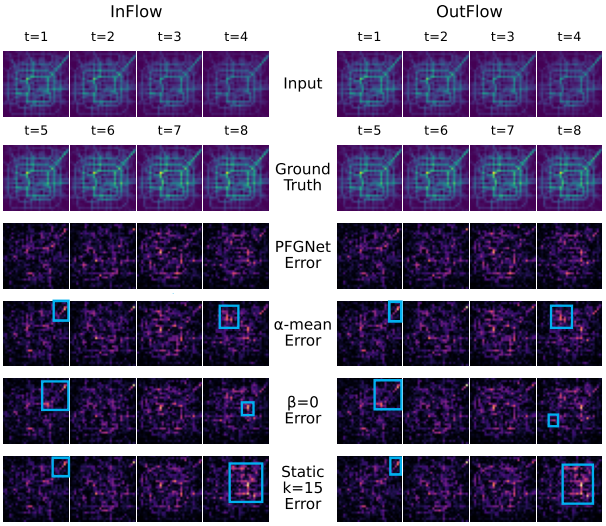


Figure 5. Qualitative visualizations on TaxiBJ.

Figure 5 presents qualitative results on TaxiBJ. We compare PFGNet error maps ($|\hat{Y} - Y|$) against three ablations: uniform fusion (α-mean), disabled center suppression ($\beta = 0$), and static kernel ($k = 15$). All maps share a unified scale. The ablations exhibit visibly larger and more diffuse errors (see highlights) compared to the sparse residuals of PFGNet. **This qualitative evidence corroborates the quantitative results in Sec. 4.3**, confirming that both adaptive gating and learnable center modulation are critical for refining structural details.

Layer-wise dissection on Human3.6M. As shown in Figure 6, we select the information-dense Human3.6M benchmark and dissect one learned PFG module, reproducing the operator behavior from the spatial to the spectral domain.

Spatial antagonistic structure (Left). The median effective kernel ($K = 31$ branch) exhibits a sign contrast between center and surround with an approximately ring-shaped tendency. This pattern is not manually imposed; it emerges from separable large-k peripheral aggregation and the learnable center–surround suppression term. The resulting shape is analogous to the classic Difference-of-Gaussians receptive-field model of retinal ganglion cells.

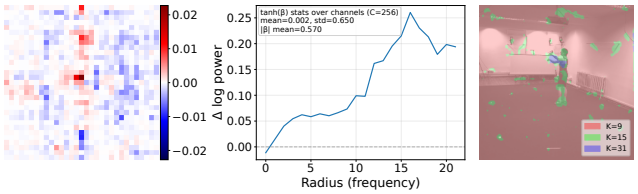


Figure 6. Mechanistic visualization of the learned PFGNet.

Adaptive spectral re-weighting (Middle). We plot the radial log-power ratio for the $K = 31$ branch against $\beta = 0$ baseline ($\Delta \log P(r) = \log P_{\text{full}} - \log P_{\beta=0}$). Energy is weakened in the low-frequency region, while strengthened in the mid-to-high frequency region to enhance motion boundaries and structural changes. The curve flattens at the highest-frequency end, indicating that high-frequency noise is not over-amplified. With the statistics in the figure, $\tanh(\beta)$ has near-zero mean and large standard deviation across channels, suggesting that the so-called suppression acts as channel-dependent bidirectional regulation: some channels suppress the center response, while others effectively compensate it. This flexibility aligns with antagonistic processing in biological vision.

Spatially adaptive kernel selection (Right). We overlay the $\arg\max$ map of the learned gating weights α, upsampled to the input resolution, on an input frame, visualizing the dominant kernel scale at each spatial location. The model prefers larger kernels around dynamic regions and motion boundaries, while assigning smaller kernels to smooth and static background, supporting input-adaptive receptive-field selection.

References

[1] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. Mau: A motion-aware unit for video prediction and beyond. *Advances in Neural Information Processing Systems*, 34:26950–26962, 2021. 4, 5

[2] 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*, 2021. 4, 5

[3] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3170–3180, 2022. 4, 5

[4] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020. 4, 5

[5] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *Computational Visual Media*, 9(4):733–752, 2023. 4, 5

[6] Xiaotao Hu, Zhewei Huang, Ailin Huang, Jun Xu, and Shuchang Zhou. A dynamic multi-scale voxel flow network for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6121–6131, 2023. 5

[7] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in Neural Information Processing Systems*, 29, 2016. 4

[8] Beibei Jin, Yu Hu, Yiming Zeng, Qiankun Tang, Shice Liu, and Jing Ye. Varnet: Exploring variations for unsupervised video prediction. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5801–5806. IEEE, 2018. 4

[9] Beibei Jin, Yu Hu, Qiankun Tang, Jingyu Niu, Zhiping Shi, Yinhe Han, and Xiaowei Li. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4554–4563, 2020. 4

[10] Nal Kalchbrenner, Aäron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *International Conference on Machine Learning*, pages 1771–1779. PMLR, 2017. 4

[11] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 4

[12] Sangmin Lee, Hak Gu Kim, Dae Hwi Choi, Hyung-Il Kim, and Yong Man Ro. Video prediction recalling long-term motion context via memory alignment learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3054–3063, 2021. 4

[13] Kunchang Li, Yali Wang, Gao Peng, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatial-temporal representation learning. In *International Conference on Learning Representations*, 2022. 4, 5

[14] Siyuan Li, Zedong Wang, Zicheng Liu, Cheng Tan, Haitao Lin, Di Wu, Zhiyuan Chen, Jiangbin Zheng, and Stan Z. Li. Moganet: Multi-order gated aggregation network. In *International Conference on Learning Representations*, 2024. 4, 5

[15] Shaohan Li, Hao Yang, Min Chen, and Xiaolin Qin. Met2net: A decoupled two-stage spatio-temporal forecasting model for complex meteorological systems. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5458–5468, 2025. 4

[16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 4, 5

[17] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 4, 5

[18] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations*, 2017. 4, 5

[19] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. In *Proceedings of the European Conference on Computer Vision*, pages 716–731, 2018. 4

[20] Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser-Lam Lim, and Jiwen Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *Advances in Neural Information Processing Systems*, 2022. 4, 5

[21] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 28, 2015. 4, 5

[22] Cheng Tan, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z. Li. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18770–18782, 2023. 4, 5

[23] Cheng Tan, Siyuan Li, Zhangyang Gao, Wenfei Guan, Zedong Wang, Zicheng Liu, Lirong Wu, and Stan Z Li. Openstl: A comprehensive benchmark of spatio-temporal predictive learning. *Advances in Neural Information Processing Systems*, 36:69819–69831, 2023. 5

[24] Cheng Tan, Zhangyang Gao, Siyuan Li, and Stan Z Li. Simvpv2: Towards simple yet powerful spatiotemporal predictive learning. *IEEE Transactions on Multimedia*, 2025. 4, 5

[25] Song Tang, Chuang Li, Pu Zhang, and RongNian Tang. Swinlstm: Improving spatiotemporal prediction accuracy using swin transformer and lstm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13470–13479, 2023. 4

[26] Yujin Tang, Peijie Dong, Zhenheng Tang, Xiaowen Chu, and Junwei Liang. Vmrnn: Integrating vision mamba and lstm for efficient and accurate spatiotemporal forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 5663–5673, 2024. 4

[27] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021. 4, 5

[28] Asher Trockman and J Zico Kolter. Patches are all you need? *Transactions on Machine Learning Research*, 2023. Featured Certification. 4, 5

[29] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in Neural Information Processing Systems*, 30, 2017. 4, 5

[30] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning*, pages 5123–5132. PMLR, 2018. 4, 5

[31] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International Conference on Learning Representations*, 2018. 4, 5

[32] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9154–9162, 2019. 4, 5

[33] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S Yu, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2208–2225, 2022. 4, 5

[34] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. In *International Conference on Learning Representations*, 2020. 4

[35] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022. 4, 5

[36] Yiqi Zhong, Luming Liang, Ilya Zharkov, and Ulrich Neumann. Mmvp: Motion-matrix-based video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4273–4283, 2023. 4