How Much Memory Do We Need? Adaptive Memory Gate for Neural Operators

Jihyeon Hur [1] **Yongseok Kwon** [1] **Min-Gi Jo** [1] **Jeongwhan Choi** [1] **Noseong Park** [1]

Abstract

Neural operators have emerged as a powerful data-driven approach for solving time-dependent PDEs. Among recent advances, memory-augmented neural operators explicitly incorporate past states and have achieved remarkable performance under low-resolution observation settings. However, existing approaches apply a fixed memory weight regardless of observation conditions, such as resolution or physical parameters, limiting their adaptability. Our preliminary experiments reveal that optimal memory weight varies with resolution and viscosity, implying that a fixed memory weight cannot simultaneously optimize performance across diverse settings. We propose AMGFNO, which dynamically modulates memory weight through a learnable gate. On the Kuramoto-Sivashinsky and Burgers' equations, AMGFNO achieves 55–79% nRMSE reduction over at low resolution, with the learned gate value automatically decreasing from $\bar{g} \approx 0.7$ to near-zero as resolution increases.

1. Introduction

Neural operators have emerged as a powerful data-driven alternative to classical numerical solvers for partial differential equations (PDEs) (Li et al., 2021; Kovachki et al., 2023a; Lu et al., 2019). Classical methods such as finite difference, finite volume, and finite element methods (LeVeque, 2007; 2002; Zienkiewicz et al., 2005) suffer from high computational cost in high dimensions and sensitivity to initial conditions. Neural operators address these limitations by learning solution operators from data, enabling fast inference after a single training phase. Among the most practically important PDE families are time-dependent PDEs, which govern physical phenomena ranging from fluid dynamics to reaction-diffusion systems (Takamoto et al., 2022). For such PDEs, neural operators commonly adopt the Markovian assumption, predicting the next state only from the current state (Tran et al., 2023; Lippe et al., 2023).

[1]KAIST, South Korea. Correspondence to: Noseong Park <noseong@kaist.ac.kr>.

Proceedings of the AI4Physics Workshop at the 43rd International Conference on Machine Learning (AI4Physics@ICML 2026), Seoul, South Korea. 2026. Copyright 2026 by the author(s).

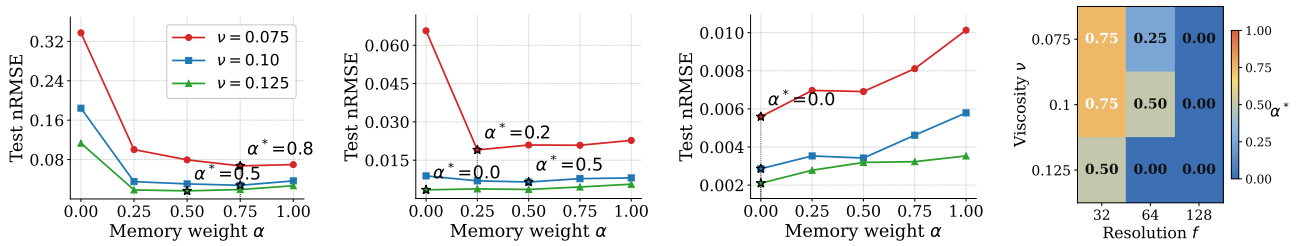
Method	Memory	Long-range	Adaptive
FFNO	✗	✗	✗
Multi-Input FFNO	✓	✗	✗
S4FFNO	✓	✓	✗
AMGFNO (Ours)	✓	✓	✓

Table 1. Comparison of neural operator architectures across three properties. Memory: incorporates past states beyond the current one; Long-range: spans the full trajectory rather than a fixed-length window; Adaptive: adjusts memory weight based on input conditions. AMGFNO is the first to achieve all three.

The Markovian assumption, however, breaks down when PDE solutions are observed at low resolution (Buitrago Ruiz et al., 2025). As resolution decreases, the Nyquist-Shannon sampling theorem limits the number of observable Fourier modes, causing significant loss of high-frequency information. Mori-Zwanzig theory formalizes partial observation as inducing strongly non-Markovian behavior, making the memory term — accumulated information from past states — decisive for accurate prediction (Parish & Duraisamy, 2017).

Fixed memory weight cannot adapt to varying conditions. Explicitly modeling memory can therefore compensate for the information lost at low resolution. Existing memory-augmented neural operators use a fixed memory weight that controls how much memory is used, regardless of observation conditions (Buitrago Ruiz et al., 2025). However, the utility of memory varies with resolution and physical parameters. Section 2.2 presents a preliminary study showing that the optimal memory weight varies with resolution and viscosity, and that no single fixed memory weight simultaneously optimizes performance across all settings.

Main idea. We propose AdaptiveMemoryGateFNO (**AMGFNO**), a neural operator that dynamically controls memory weight via a adaptive memory gate. As summarized in Table 1, FFNO uses no memory, Multi-Input FFNO captures only short-range history, and S4FFNO (Buitrago Ruiz et al., 2025) introduces long-range memory but lacks adaptive control. AMGFNO is the first to combine all three prop-



(a) $f = 32$ *(b)* $f = 64$ *(c)* $f = 128$ *(d)* Optimal α^*

Figure 1. Effect of memory weight α on nRMSE across resolutions f and viscosity ν values on the KS equation. Panels (a-c) show nRMSE as a function of α for three viscosity values ($\nu \in \{0.075, 0.10, 0.125\}$) at a fixed resolution, with stars indicating optimal memory weight α^*. Panel (d) summarizes the optimal memory weight α^* across all settings.

erties: (1) using past states as memory beyond the current timestep, (2) capturing long-range temporal dependencies across the full trajectory rather than a fixed window, and (3) adaptively adjusting memory weight based on the current input condition rather than using a fixed weight. Throughout this paper, we use α to denote the manually-tuned memory weight in our ablation study, α^* for the optimal memory weight that minimizes error per condition, and $\bar{g}$ for AMGFNO's learned mean memory gate value. The contributions of AMGFNO are summarized as follows:

- **Motivation.** We empirically show that the optimal memory weight α^* varies with resolution and viscosity (Section 2.2), and theoretically show that the gain of memory can converge to zero in the high-resolution limit (Section 2.3).

- **Method.** We propose AMGFNO (Section 3), which dynamically modulates memory weight via an adaptive memory gate. The learned memory gate value decreases monotonically from $\bar{g} \approx 0.7$ to near-zero as resolution increases, consistent with theoretical predictions.

- **Results.** On the KS equation across three viscosity values and three resolutions, as well as Burgers' equation, AMGFNO achieves 55–79% nRMSE reduction over the best fixed-memory baseline (S4FFNO) at low resolution ($f = 32$) and 25–40% at high resolution ($f = 128$), demonstrating effective memory adaptation across all experimental settings (Section 4).

2. Preliminaries and Motivation

2.1. Time-dependent PDE Problem Definition

We consider a time-dependent partial differential equation defined on a spatial domain $\Omega \subset \mathbb{R}^d$ and a time interval $[0, T]$:

$$\frac{\partial u}{\partial t}(t, x) = \mathcal{L}_\rho[u](t, x), \qquad (t, x) \in [0, T] \times \Omega. \tag{1}$$

The PDE is equipped with an initial condition

$$u(0, x) = u_0(x), \qquad x \in \Omega, \tag{2}$$

and a boundary condition

$$\mathcal{B}[u|_{\partial\Omega}](t) = 0, \qquad t \in [0, T]. \tag{3}$$

Here $u(t, \cdot) : \Omega \to \mathbb{R}^V$ denotes the PDE state at time t, x is the spatial coordinate, V is the channel dimension of the input feature space, $\mathcal{L}_\rho$ is a differential operator in x parameterized by physical parameters ρ, and $\mathcal{B}$ specifies the boundary condition. In our one-dimensional experiments, we use periodic boundary conditions. Let Φ_t denote the solution map induced by the PDE, so that

$$u(t, \cdot) = \Phi_t(u_0).$$

The learning problem considered in this paper is to approximate the time evolution of this solution map from discretized observations of the PDE state. The resolved-state and spectral-truncation notation used in the theory is introduced in Section 2.3. Dataset-specific PDE equations, parameter values, and data-generation details are deferred to Appendix F.

2.2. Preliminary Study: How Much Memory Should Be Used?

To analyze when memory is beneficial, we conduct an ablation study that manually controls the memory weight in S4FFNO (Buitrago Ruiz et al., 2025). S4FFNO follows a FFNO–FFNO–S4–FFNO–FFNO layer structure, where the S4 memory layer is inserted at the middle position ($\ell^* = 3$). Let $\mathbf{h}_t^{(\ell^*-1)} \in \mathbb{R}^{N \times H}$ denote the hidden state at the output of the second FFNO layer, and let $\mathbf{z} = \mathcal{M}(\mathbf{u}_t) \in \mathbb{R}^{N \times H}$ denote the output of the S4 memory branch. We replace the original residual fusion with a convex combination that explicitly controls the relative weight of memory:

$$\mathbf{h}_{t+1} \leftarrow \alpha \cdot \mathbf{z} + (1 - \alpha) \cdot \mathbf{h}_t, \qquad \alpha \in [0, 1], \tag{4}$$

where the updated $\mathbf{h}_{t+1}$ is passed to the third FFNO layer. Here, α is the memory weight: $\alpha = 0$ corresponds

to a pure Markovian model that bypasses the memory branch, $\alpha = 1$ fully replaces the residual with the memory output. We sweep $\alpha \in \{0, 0.25, 0.5, 0.75, 1.0\}$ on the Kuramoto–Sivashinsky (KS) equation across three spatial resolutions ($f \in \{32, 64, 128\}$) and three viscosity values ($\nu \in \{0.075, 0.10, 0.125\}$). Results are shown in Figure 1. The same observations hold under an additive parameterization $\mathbf{h}_{t+1} \leftarrow \alpha \cdot \mathbf{z} + \mathbf{h}_t$; see Appendix A for details.

Observation 1: Optimal memory weight varies significantly across conditions. Figure 1(a-c) shows test nRMSE as a function of α for different resolutions and viscosities, with stars marking the optimal memory weight α^*. At low resolution ($f = 32$), using optimal $\alpha^* \in \{0.5, 0.75\}$ reduces error by up to 41% compared to pure memory ($\alpha = 1$): from 0.070 to 0.067 at $\nu = 0.075$, from 0.037 to 0.028 at $\nu = 0.10$, and from 0.027 to 0.016 at $\nu = 0.125$. At high resolution ($f = 128$), $\alpha^* = 0$ across all viscosities. At intermediate resolution ($f = 64$), α^* varies from 0 to 0.5 depending on viscosity. As the heatmap in Figure 1(d) shows, optimal α^* spans $\{0.0, 0.25, 0.5, 0.75\}$ across the 9 settings—no single fixed value is optimal, motivating adaptive memory gate. This trend holds under both convex and additive fusion (Appendix A).

Observation 2: Fixed α fails to generalize across setting. As the heatmap in Figure 1(d) shows, a fixed α optimized for one condition performs poorly in others. For instance, $\alpha = 0.75$ works well at $f = 32$ (high viscosity) but degrades performance at $f = 128$, while $\alpha = 0$ underexploits memory at low resolution. At intermediate resolution ($f = 64$, Figure 1(b)), optimal α^* varies widely—from 0 to 0.5 across viscosities—illustrating that fine-grained, condition-dependent control is necessary.

Motivation. These findings reveal that memory weight should adapt dynamically to the simulation condition. A fixed α treats memory as a static architectural choice, ignoring that optimal memory usage depends on the observation setting (e.g., resolution, viscosity). We propose an Adaptive Memory Gate Block in Section 3.3 that learns to modulate memory based on the current hidden state, enabling the model to automatically adjust across different observation settings without manual tuning.

2.3. Theoretical Motivation

Intuition. Memory should be treated as a condition-dependent resource. At low observation resolution, unresolved Fourier modes may still influence the next resolved state, making the observed dynamics effectively non-Markovian and temporal history informative. As the resolution increases, this projection-induced hidden influence diminishes; a fixed memory path then has decreasing approximation benefit while still contributing additional finite-sample complexity.

Setup. We formalize this mechanism under the ideal Fourier projection analyzed in Appendix G. Let $m_f := \lfloor f/2 \rfloor$, let P_f denote projection onto Fourier modes $|n| \le m_f$, and let $Q_f = I - P_f$ denote the unresolved tail. For a trajectory $u^\rho_{t+1} = \Phi^\rho_{\Delta t}(u^\rho_t)$ under physical condition ρ, write

$$u^\rho_{t,f} := P_f u^\rho_t, \quad Y^{f,\rho}_t := P_f u^\rho_{t+1},$$
$$\mathcal{U}^{f,\rho}_t := (u^\rho_{0,f}, \dots, u^\rho_{t,f}).$$

Here $u^\rho_{t,f}$ is the resolved state, $\mathcal{U}^{f,\rho}_t$ is the resolved history, and $Y^{f,\rho}_t$ is the one-step resolved target. The influence of the unresolved tail $Q_f u^\rho_t$ on the next resolved state is measured by

$$\delta_{t,f,\rho} := P_f \Phi^\rho_{\Delta t}(u^\rho_t) - P_f \Phi^\rho_{\Delta t}(P_f u^\rho_t). \tag{5}$$

Under the finite-horizon regularity and local Lipschitz assumptions in Appendix G,

$$\mathbb{E}\|\delta_{t,f,\rho}\|^2 \le \kappa_{\text{tail},\rho} f^{-2\beta}. \tag{6}$$

Thus, as f grows, the resolved dynamics become increasingly close to a Markovian system in the current resolved state.

From residual decay to memory-gain bounds. To connect this residual decay to neural-operator approximation, let $\mathcal{T}_{f,\rho}(\phi) = P_f \Phi^\rho_{\Delta t}(\phi)$ denote the projected Markovian target, and let $b_{f,\rho}(\mathcal{U}^{f,\rho}_t)$ denote the history-conditioned Bayes target. We write $\mathcal{E}^{\mathcal{C}}_{f,\rho}$ for the approximation error of a model class $\mathcal{C}$ to this Bayes target, and $\mathcal{A}^F_{f,\rho}$ for FFNO's approximation error to the projected Markovian target. Formal definitions are given in Appendix G.

The distinction between $\mathcal{A}^F_{f,\rho}$ and $\mathcal{E}^F_{f,\rho}$ is important. The assumption is imposed only on the projected Markovian target $\mathcal{T}_{f,\rho}$ through $\mathcal{A}^F_{f,\rho}$; it does not assume that FFNO can approximate the history-conditioned Bayes target. Theorem 2.1 uses the residual decay in Eq. (6) to transfer projected-target approximation into a bound on the actual Bayes-target error $\mathcal{E}^F_{f,\rho}$.

Define the memory gains over FFNO by

$$\Gamma^{S4}_{f,\rho} := \mathcal{E}^F_{f,\rho} - \mathcal{E}^{S4}_{f,\rho}, \qquad \Gamma^{AMG}_{f,\rho}(\lambda) := \mathcal{E}^F_{f,\rho} - \mathcal{E}^{AMG}_{f,\rho}(\lambda). \tag{7}$$

Theorem 2.1 (Convergence of projection-induced memory gain)**.** *Under the assumptions G.3–G.5,*

$$\mathcal{E}^F_{f,\rho} \le 2\mathcal{A}^F_{f,\rho} + 2\kappa_{\text{tail},\rho} f^{-2\beta}. \tag{8}$$

Consequently, $\mathcal{E}^F_{f,\rho} \to 0$ as $f \to \infty$. Since the memory-augmented classes contain the FFNO fallback,

$$0 \le \Gamma^{S4}_{f,\rho} \le \mathcal{E}^F_{f,\rho}, \qquad 0 \le \Gamma^{AMG}_{f,\rho}(\lambda) \le \mathcal{E}^F_{f,\rho} \tag{9}$$

for every $\lambda \geq 0$. Hence the additional approximation gain of memory induced by spectral projection vanishes in the high-resolution limit.

The proof is given in Appendix G.4.

AMGFNO is designed to track this convergence through its frequency-aware gate. For a full reference state ϕ with Fourier coefficients $a_n(\phi)$, define the unresolved spectral-energy ratio

$$\omega_f(\phi) := \frac{\sum_{|n|>m_f} |a_n(\phi)|^2}{\sum_{n\in\mathbb{Z}} |a_n(\phi)|^2}, \tag{10}$$

with value zero when the denominator is zero. For condition (f, ρ), define the population unresolved-energy ratio

$$\omega_{f,\rho}^{\text{pop}} := \mathbb{E}_{u_0,t}\left[\omega_f(u_t^\rho)\right]. \tag{11}$$

In experiments, $\omega_{f,\rho}^{\text{pop}}$ is replaced by the empirical training-set estimate $\widehat{\omega}_{f,\rho}$, computed from high-resolution training trajectories and held fixed for the condition (f, ρ). This quantity is used only as a condition-level calibration constant; no high-resolution test information is used.

The scalar frequency gate is

$$\chi_{f,\rho} := \begin{cases} \sigma\Big(w_1 \log \omega_{f,\rho}^{\text{pop}} + b_\omega\Big), & \omega_{f,\rho}^{\text{pop}} > 0, \\ 0, & \omega_{f,\rho}^{\text{pop}} = 0, \end{cases} \qquad w_1 > 0. \tag{12}$$

Lemma G.10 shows that $\omega_{f,\rho}^{\text{pop}} \to 0$, and therefore $\chi_{f,\rho} \to 0$. Since the content gate is bounded entrywise by one, the induced AMGFNO memory budget is bounded by

$$\lambda_{f,\rho} := \chi_{f,\rho}^2, \tag{13}$$

which also converges to zero.

Finally, the adaptive memory gate also controls finite-sample complexity. Let $R_m(\mathfrak{L})$ denote the Rademacher complexity of a loss class, and let $U_{f,\rho}^F$, $U_{f,\rho}^{S4}$, and $U_{f,\rho}^{AMG}(\lambda)$ denote the ERM upper bounds defined in Appendix G.6.

Theorem 2.2 (Adaptive high-resolution comparison)**.** *Let $\lambda_{f,\rho} = \chi_{f,\rho}^2$. Under the assumptions G.3–G.7 and the sufficient conditions of Proposition G.12,*

$$R_m(\mathfrak{L}_{f,\rho}^{AMG}(\lambda_{f,\rho})) - R_m(\mathfrak{L}_{f,\rho}^{F}) = O(\sqrt{\lambda_{f,\rho}}) \to 0. \tag{14}$$

Thus,

$$U_{f,\rho}^{AMG}(\lambda_{f,\rho};\eta) - U_{f,\rho}^{F}(\eta) \to 0. \tag{15}$$

If fixed-fusion S4FFNO retains a positive asymptotic Rademacher-complexity gap over FFNO, then for all sufficiently large f,

$$U_{f,\rho}^{AMG}(\lambda_{f,\rho};\eta) < U_{f,\rho}^{S4}(\eta). \tag{16}$$

The proof is given in Appendix G.8.

The analysis isolates the memory benefit induced by projection-based loss of Fourier modes. It does not rule out persistent memory gains from other sources of partial observability, such as latent parameters, stochastic forcing, measurement noise, or temporal subsampling. The theory controls one-step resolved prediction risk, while accumulated autoregressive rollout error is evaluated empirically in Section 4.

3. Proposed Method

AMGFNO builds on the FFNO backbone by inserting an Adaptive Memory Gate (AMG) Block at the middle layer, as illustrated in Figure 2. The AMG Block consists of a memory layer and an adaptive memory gate (AMG). The AMG consists of content gate and frequency-aware gate. The input PDE state $\mathbf{u}_t \in \mathbb{R}^{S\times V}$ is first encoded into a latent representation $\mathbf{h}_t^{(0)} = \text{Enc}(\mathbf{u}_t, \mathbf{s}) \in \mathbb{R}^{S\times H}$, where $\mathbf{s} \in \mathbb{R}^{S\times 1}$ is the spatial grid and H is the hidden dimension. The latent state then passes through $L = 5$ layers in the order FFNO−FFNO−AMG−FFNO−FFNO:

$$\mathbf{h}_t^{(\ell)} = \begin{cases} \mathcal{F}_\ell\Big(\mathbf{h}_t^{(\ell-1)}\Big) + \mathbf{h}_t^{(\ell-1)}, & \ell \neq \ell^* \\ \mathbf{g}_t \odot \mathbf{z}_t + (1-\mathbf{g}_t)\odot \mathbf{h}_t^{(\ell-1)}, & \ell = \ell^* \end{cases} \tag{17}$$

where $\ell^* = 3$ is the index of the Adaptive Memory Gate (AMG) Block and $\mathbf{z}_t = \mathcal{M}(\mathbf{h}_{1:t}^{(\ell^*-1)})$ is the memory layer output. At $\ell \neq \ell^*$, each layer applies an FFNO operation with a residual connection. At $\ell = \ell^*$, the AMG Block combines the memory output $\mathbf{z}_t$ and the Markovian hidden state $\mathbf{h}_t^{(\ell^*-1)}$ via the adaptive memory gate $\mathbf{g}_t \in [0,1]^{S\times H}$, which is computed from the current feature state. The gate dynamically adjusts memory weight across different simulation conditions, allowing the model to modulate memory usage based on the learned representation. The final prediction is $\hat{\mathbf{u}}_{t+1} = \text{Dec}(\mathbf{h}_t^{(L)})$.

3.1. Factorized Fourier Neural Operator (FFNO) Layer

Each FFNO layer $\mathcal{F}_\ell$ operates along the *spatial* axis and combines a spectral path with a local linear path:

$$\mathcal{F}_\ell(\mathbf{h}) = \mathcal{F}^{-1}\Big(\mathbf{W}_\ell^{\text{freq}} \cdot [\mathcal{F}(\mathbf{h})]_{1:K}\Big) + \mathbf{W}_\ell^{\text{loc}}\,\mathbf{h}, \tag{18}$$

where $\mathcal{F}$ is the discrete Fourier transform along the spatial dimension, K is the number of retained Fourier modes, and $\mathbf{W}_\ell^{\text{freq}}$, $\mathbf{W}_\ell^{\text{loc}}$ are learnable weight matrices.

3.2. Memory Layer

The memory output $\mathbf{z}_t = \mathcal{M}(\mathbf{h}_{1:t})$ is computed by applying S4 (Structured State Space Model) along the *temporal* axis.

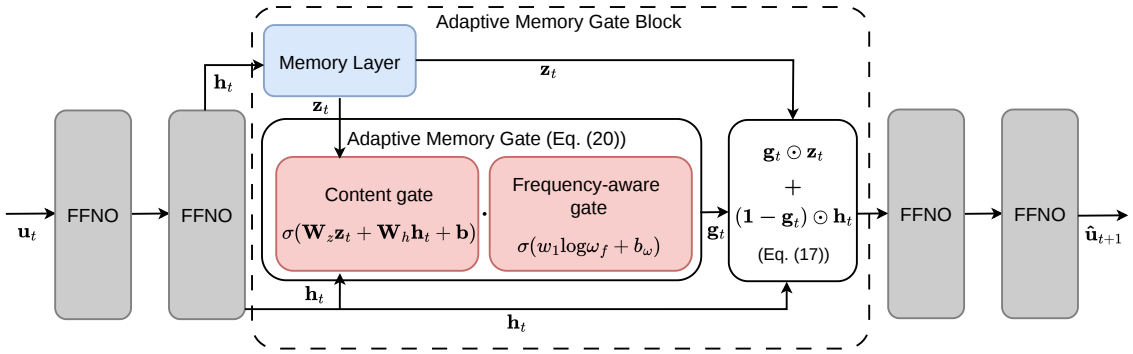


Figure 2. Overview of the AMGFNO architecture. The model consists of five layers with an Adaptive Memory Gate (AMG) Block at the middle layer ($\ell = 3$). The AMG Block combines the memory layer output $\mathbf{z}_t$ and the Markovian state $\mathbf{h}_t$ (output of the second FFNO layer, $\ell = 2$) using an adaptive memory gate $\mathbf{g}_t$ that is decomposed into two components: a content gate $\sigma(\mathbf{W}_z\mathbf{z}_t + \mathbf{W}_h\mathbf{h}_t + \mathbf{b})$ for input-adaptive control and a frequency-aware gate $\sigma(w_1 \log \omega_f + b_\omega)$ for conditional priors. The final output is computed as $\mathbf{g}_t \odot \mathbf{z}_t + (1 - \mathbf{g}_t) \odot \mathbf{h}_t$.

For each spatial position $p \in \{1, \ldots, S\}$ independently:

$$\mathbf{v}_t^{(p)} = \mathbf{A}\,\mathbf{v}_{t-1}^{(p)} + \mathbf{B}\,\mathbf{h}_t^{(p)}, \qquad \mathbf{z}_t^{(p)} = \mathbf{C}\,\mathbf{v}_t^{(p)}, \tag{19}$$

where $\mathbf{v}_t^{(p)} \in \mathbb{R}^N$ is the S4 internal state that accumulates temporal context up to time t, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are learnable parameters initialized with HiPPO (Gu et al., 2020). The spatial dimension is treated as a batch axis, so all positions share the same SSM parameters.

3.3. Adaptive Memory Gate

As shown in Figure 2, the adaptive memory gate $\mathbf{g}_t \in (0,1)^{S\times H}$ controls how much the memory output $\mathbf{z}_t$ contributes relative to the current hidden state $\mathbf{h}_t$. The gate consists of a content gate and a frequency-aware gate:

$$\mathbf{g}_t = \underbrace{\sigma(\mathbf{W}_z\,\mathbf{z}_t + \mathbf{W}_h\,\mathbf{h}_t + \mathbf{b})}_{\text{content gate} \in (0,1)^{S\times H}} \cdot \underbrace{\sigma(w_1 \log \omega_f + b_\omega)}_{\text{frequency-aware gate} \in (0,1)}. \tag{20}$$

Content gate. The content gate $\sigma(\mathbf{W}_z\mathbf{z}_t + \mathbf{W}_h\mathbf{h}_t + \mathbf{b}) \in (0,1)^{S\times H}$ provides input-adaptive control that varies per sample, channel, and spatial location. It takes the memory output $\mathbf{z}_t$ and the current hidden state $\mathbf{h}_t$ as input and learns to modulate memory weight based on their learned representations. The parameters $\mathbf{W}_z$, $\mathbf{W}_h$, and $\mathbf{b}$ are initialized to zero, allowing the content gate to learn data-driven corrections during training while deferring to the frequency-aware prior at initialization.

Frequency-aware gate. The frequency-aware gate $\sigma(w_1 \log \omega_f + b_\omega) \in (0,1)$ supplies a conditional prior based on the observation condition (f, ρ). Here ω_f is the high-frequency energy ratio defined in Eq. (11), computed as the mean over the training set and held fixed during training and inference. We found empirically that this fixed ω_f estimate consistently outperforms per-batch computation by 5–20% across settings (see Appendix B), suggesting that ω_f serves best as a calibration constant characterizing the observation condition rather than a dynamic per-sample feature. The parameters $w_1 = 2.0$ and $b_\omega = 3.8$ are analytically initialized from two anchor conditions (ω_{32} and ω_{128}) to provide a resolution-aware prior that the content gate refines during training.

4. Experiments

4.1. Experimental Setup

Benchmarks and observation settings. We evaluate on two 1D time-dependent PDEs with periodic boundary conditions. For the Kuramoto–Sivashinsky (KS) equation, we consider three viscosity values $\nu \in \{0.075, 0.1, 0.125\}$ to span different spectral conditions. Lower viscosity leads to more chaotic dynamics with significant high-frequency content, while higher viscosity smooths the solution. For Burgers' equation, we use viscosity $\nu = 0.001$, which maintains strong nonlinear advection. To assess how memory usage should adapt across observation conditions, we consider three spatial resolutions $f \in \{32, 64, 128\}$, representing low-, medium-, and high-resolution, respectively. At low resolution, many high-frequency Fourier modes are unobserved; at high resolution, most spectral information is directly observable. PDE-specific parameters, final simulation times, and data generation procedures are detailed in Appendix F.

Training objective and procedure. All models are trained with a one-step prediction objective. Given the current resolved state $u_j^{f,(i)}$ on spatial grid S_f at time t_j, the

Architecture	Uses memory	Kuramoto–Sivashinsky									Burgers'		
		$f = 32$			$f = 64$			$f = 128$			$f = 32$	$f = 64$	$f = 128$
		ν=.075	ν=.10	ν=.125	ν=.075	ν=.10	ν=.125	ν=.075	ν=.10	ν=.125			
Factformer (1D)	✗	0.436	0.391	0.149	0.195	0.086	0.022	0.058	0.030	0.017	0.190	0.162	0.117
GKT	✗	0.588	0.601	0.314	0.401	0.120	0.016	0.028	0.013	0.007	0.356	0.349	0.307
U-Net	✗	0.542	0.511	0.249	0.147	0.062	0.022	0.033	0.027	0.014	0.188	0.171	0.112
FFNO	✗	0.500	0.446	0.187	0.107	0.033	0.004	**0.006**	0.004	**0.002**	0.207	0.146	0.099
Multi Input FFNO	✓	0.364	0.308	0.092	0.108	0.046	0.005	0.057	0.052	0.023	0.099	0.054	0.028
S4FFNO	✓	0.139	0.108	0.031	0.036	0.011	0.004	0.008	0.005	0.003	0.053	0.037	0.030
AMGFNO (Ours)	✓	**0.062**	**0.023**	**0.014**	**0.021**	**0.007**	**0.003**	**0.006**	**0.003**	**0.002**	**0.024**	**0.018**	**0.020**

Table 2. nRMSE values at different resolutions for KS and Burgers' equations. AMGFNO consistently outperforms all baselines across resolutions, achieving up to 79% nRMSE reduction over S4FFNO at $f = 32$ while suppressing memory at $f = 128$ to match or surpass FFNO.

Table 3. Converged mean gate value $\bar{g}$ (mean of last 20 epochs) for AMGFNO across different resolution and viscosity settings.

Dataset	ν	$\bar{g}$ by Resolution f		
		$f = 32$	$f = 64$	$f = 128$
KS	0.075	0.6915	0.4720	0.0030
	0.10	0.7036	0.2150	0.0008
	0.125	0.6976	0.0721	0.0002
Burgers'	0.001	0.0267	0.0165	0.0020

model predicts the next state $u_{j+1}^{f,(i)}$ at time t_{j+1}. During training, we employ teacher forcing: each step receives the ground-truth state as input rather than the model's own prediction from the previous step. For memory-augmented models, the internal memory state is updated using the same teacher-forced sequence. The training loss is the mean squared error averaged over mini-batches $\mathcal{B}$ and all timesteps within each trajectory:

$$\mathcal{L}_{\text{train}}(\theta) = \frac{1}{|\mathcal{B}|J} \sum_{i \in \mathcal{B}} \sum_{j=0}^{J-1} \left\| \mathcal{G}_\theta(u_j^{f,(i)}, g_f) - u_{j+1}^{f,(i)} \right\|_2^2, \tag{21}$$

where $\mathcal{G}_\theta$ is the neural operator, g_f is the grid encoding, and J is the number of timesteps per trajectory. All models use the same optimizer settings and are trained for 200 epochs.

Evaluation protocol. At test time, models are evaluated via autoregressive rollout. Only the initial state $u_0^{f,(i)}$ is provided as ground truth; subsequent predictions are generated by repeatedly feeding the model's own output back as input:

$$\begin{aligned} \widetilde{u}_0^{f,(i)} &= u_0^{f,(i)}, \\ \widetilde{u}_{j+1}^{f,(i)} &= \mathcal{G}_\theta(\widetilde{u}_j^{f,(i)}, g_f), \quad j = 0, \ldots, J-1. \end{aligned} \tag{22}$$

This autoregressive evaluation measures the model's ability to maintain accurate predictions over long rollouts, where one-step errors can accumulate. Performance is measured by the normalized root mean squared error (nRMSE), averaged over all test trajectories and rollout timesteps:

$$\text{nRMSE}(f) = \frac{1}{N_{\text{test}} J} \sum_{i=1}^{N_{\text{test}}} \sum_{j=1}^{J} \frac{\left\| \widetilde{u}_j^{f,(i)} - u_j^{f,(i)} \right\|_2}{\left\| u_j^{f,(i)} \right\|_2}, \tag{23}$$

where N_{test} is the number of test trajectories. Lower nRMSE indicates more accurate trajectory prediction.

Baseline comparisons. We compare AMGFNO against two categories of baselines. Markovian neural operators use only the current state without temporal memory: Factformer (Li et al., 2023b), Galerkin Transformer(GKT) (Cao, 2021), U-Net (Gupta & Brandstetter, 2023),and FFNO (Tran et al., 2023). Memory-augmented operators incorporate information from past states: Multi-Input FFNO (Buitrago Ruiz et al., 2025), which concatenates the previous four timesteps as input, and S4FFNO (Buitrago Ruiz et al., 2025), which uses a state-space memory layer with fixed fusion. All baselines use comparable model capacity and are trained with identical hyperparameters for fair comparison. Architectural details and implementation specifics are provided in Appendix F.

4.2. Main Result

Table 2 shows nRMSE results for the KS equation across three viscosity values and three resolutions, as well as for Burgers' equation. AMGFNO consistently outperforms all baselines across all resolutions and viscosities, with the largest performance gap at low resolution and the smallest at high resolution.

Low resolution ($f = 32$). AMGFNO significantly outperforms all baselines across all viscosities at resolution 32. Compared to the previous best-performing model S4FFNO, nRMSE on KS decreases from 0.139 to 0.062 at $\nu = 0.075$, from 0.108 to 0.023 at $\nu = 0.1$, and from 0.031 to 0.014 at $\nu = 0.125$. On Burgers', nRMSE decreases from 0.053 to 0.024 compared to S4FFNO.

AMGFNO improves over baselines by adaptively adjusting

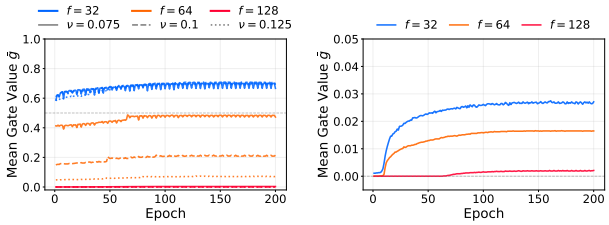
the blend between the current state and temporal memory. At low resolution, the gate increases memory weight to compensate for unobserved high-frequency modes, whereas S4FFNO uses a fixed blend ratio regardless of the observation condition. As shown in Table 3 at resolution 32, the $\bar{g}$ values are larger compared to resolutions 64 and 128. Since $\bar{g}$ represents the degree of memory usage via the adaptive memory gate, this confirms that the gate adjusts to increase memory usage at low resolution. For KS, $\bar{g} \approx 0.70$, and for Burgers', $\bar{g} = 0.0267$, which is larger than the 0.0165 observed at resolution 64.

Medium resolution ($f = 64$). AMGFNO outperforms all baselines across all viscosities at resolution 64. Compared to the previous best, S4FFNO, nRMSE on KS decreases from 0.036 to 0.021 at $\nu = 0.075$, from 0.011 to 0.007 at $\nu = 0.1$, and from 0.004 to 0.003 at $\nu = 0.125$. On Burgers', nRMSE decreases from 0.037 to 0.018 compared to S4FFNO. At resolution 64, the adaptive memory gate $\bar{g}$ varies substantially with viscosity, adjusting memory usage appropriately. At $\nu = 0.075$, $\bar{g} = 0.4720$, meaning approximately 47% of the memory path and 53% of the Markovian path are utilized as shown in Figure 2. At $\nu = 0.125$, $\bar{g} = 0.0721$, meaning the memory path contributes only about 7% while the Markovian path contributes about 93%; that is, at $\nu = 0.125$ the adaptive memory gate partially closes with a low $\bar{g}$ value. This is consistent with the finding in Section 2.2 that optimal memory weight depends on viscosity: Table 3 shows that the adaptive memory gate $\bar{g}$ modulates accordingly across different ν values.

High resolution ($f = 128$). At high resolution, the adaptive memory gate value is nearly closed—0.003 at $\nu = 0.075$, 0.0008 at $\nu = 0.1$, and 0.0002 at $\nu = 0.125$—effectively guiding AMGFNO into the Markovian condition. S4FFNO uses fixed memory and at resolution 128 actually performs slightly worse than FFNO: at $\nu = 0.075$, S4FFNO's nRMSE increases from 0.006 to 0.008 compared to FFNO, indicating that memory usage is harmful at high resolution. AMGFNO addresses this limitation by incorporating an adaptive memory gate to regulate memory usage, achieving performance comparable to or better than FFNO across all viscosities on KS. On Burgers', AMGFNO achieves the lowest nRMSE of 0.020, compared to 0.099 for FFNO, 0.028 for Multi Input FFNO, and 0.030 for S4FFNO.

4.3. Additional Analysis

Resolution and viscosity-dependent adaptation. As shown in Table 3, the learned gate values adapt to both resolution and viscosity. For KS, the gate actively leverages memory at $f = 32$ with $\bar{g} \approx 0.70$, then nearly closes at $f = 128$ with $\bar{g} \leq 0.003$. At medium resolution ($f = 64$), viscosity-dependent variation is pronounced: $\bar{g} = 0.472$ at $\nu = 0.075$ versus $\bar{g} = 0.072$ at $\nu = 0.125$, demonstrating that the gate selectively opens in low-viscosity settings where high-frequency components remain significant and closes when the solution is well captured at the observation resolution. For Burgers', gate values are relatively small across all resolutions ($\bar{g} \leq 0.027$), reflecting the PDE's intrinsic characteristic where the viscous term νu_{xx} acts as a low-pass filter that monotonically attenuates high-frequency components.



(a) KS *(b)* Burgers ($\nu = 0.001$)

Figure 3. Adaptive memory gate behavior. Mean gate value $\bar{g}$ during training across resolutions f.

Gate evolution during training. Figure 3 shows gate value evolution over epochs. For the KS equation (left panel), the gate values are initialized differently by resolution and viscosity, showing distinct patterns from epoch 0. At low resolution ($f = 32$), the gates start large ($\bar{g} \approx 0.60$–0.70) across all viscosities. At medium resolution ($f = 64$), viscosity-dependent initialization emerges: $\bar{g} \approx 0.47$ for $\nu = 0.075$, $\bar{g} \approx 0.21$ for $\nu = 0.1$, and $\bar{g} \approx 0.07$ for $\nu = 0.125$. At high resolution ($f = 128$), the gates are initialized near zero regardless of viscosity. During training, these initial values are fine-tuned incrementally.

The right panel shows gate evolution for Burgers' equation at $\nu = 0.001$. At epoch 0, all gate values start near zero regardless of resolution. As training progresses, the gate values separate by resolution: the low-resolution gate increases substantially, while the high-resolution gate remains near zero. This confirms that gate values are automatically learned without manual tuning per condition. Appendix E shows the co-evolution of gate values and training loss across all conditions.

Role of the frequency-aware gate. In our implementation, we optionally include a frequency-aware gate $\sigma(w_1 \log \omega_f + b_\omega)$ in addition to the content gate, which provides a conditional prior at initialization. While the content gate alone can learn adaptive memory modulation across most settings, an ablation study (Appendix C) shows that the frequency-aware gate is most beneficial at high resolution in KS, where $\omega_f \to 0$ and it correctly suppresses unnecessary memory usage.

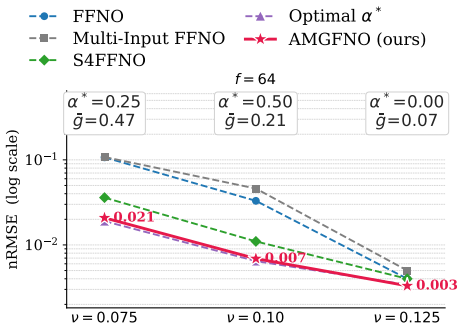


Figure 4. Performance comparison on the Kuramoto-Sivashinsky (KS) equation at resolution 64. FFNO, Multi-Input FFNO, S4FFNO baselines and optimal memory weight (α^*) are compared to the proposed AMGFNO across different viscosities ($\nu \in \{0.075, 0.1, 0.125\}$). $\bar{g}$ denotes the adaptive memory gate value learned by AMGFNO.

Comparison with Optimal Fixed Scaling Figure 4 demonstrates the necessity of adaptive gating by comparing against the optimal fixed memory weight α^* from Section 2.2 at resolution $f = 64$ on the Kuramoto-Sivashinsky (KS) equation. The optimal α^* varies across KS viscosities: $\alpha^* = 0.25$ at $\nu = 0.075$, $\alpha^* = 0.50$ at $\nu = 0.10$, and $\alpha^* = 0.00$ at $\nu = 0.125$, confirming that no single fixed value works across conditions. AMGFNO's learned gate values $\bar{g} = 0.47$, 0.21, and 0.07 also adapt across these viscosity conditions, achieving nRMSE of 0.021, 0.007, and 0.003 that closely matches the performance of the manually-tuned optimal at each condition. Crucially, while finding α^* requires exhaustive search per condition, AMGFNO discovers the appropriate memory weight automatically through training on the KS dataset. Results for resolutions 32 and 128 are in Appendix C.

5. Related Work

Neural operators for PDEs. The Fourier Neural Operator (FNO) (Li et al., 2021) and its factorized variant FFNO (Tran et al., 2023) have become standard Markovian baselines for PDE operator learning. Other Markovian operators explore alternative inductive biases, including multi-scale spatial structure (Gupta & Brandstetter, 2023), kernel-based attention (Cao, 2021), and transformer-style architectures with factorized or axial attention (Li et al., 2023a;b). Despite their architectural diversity, these models share the Markovian assumption, predicting the next state from the current one alone — a design that becomes inadequate when the current state does not fully resolve the underlying dynamics.

Memory in PDE models. Memory-augmented neural operators incorporate past states as additional input to relax the Markovian assumption. The earliest such variant, Multi-Input FFNO, simply concatenates a short fixed-length window of past states as additional input channels, providing only short-range, uncompressed memory. MemNO (Buitrago Ruiz et al., 2025) extends this idea by replacing the input window with an S4 layer (Gu et al., 2022) that compresses the entire past trajectory along the time axis, and provides theoretical and empirical evidence that such long-range memory is beneficial under partial observation. Concretely, MemNO inserts a single S4 layer between Markovian FFNO layers and adds the resulting temporal-memory representation back via a residual connection. In both designs, however, the memory weight is fixed regardless of the input — yet, as we show, the optimal memory weight depends on observation conditions.

Gating mechanisms. The variation in optimal memory weight points to a need for input-dependent control of memory usage. Such control is well established in sequence modeling, where multiplicative gates have a long history — from LSTM (Hochreiter & Schmidhuber, 1997) and GRU (Cho et al., 2014) to Highway Networks (Srivastava et al., 2015) and selective state-space models such as Mamba (Gu & Dao, 2024); in all cases, gates regulate information flow as a function of the input. In neural operators, by contrast, memory has been integrated by a fixed, input-independent addition (Buitrago Ruiz et al., 2025).

Position of our work. AMGFNO bridges these two lines of work. We inherit the temporal-memory architecture of MemNO but replace its fixed fusion with an adaptive gate that combines a content gate and frequency-aware gate. Unlike prior memory-augmented neural operators, AMGFNO modulates modulates memory weight through a learnable gate.

6. Conclusion and Future Work

We demonstrate the necessity of adaptive memory control in neural operators for time-dependent PDEs. Preliminary studies reveal that optimal memory weight varies across observation conditions. AMGFNO incorporates an adaptive memory gate that learns to modulate memory usage based on the current condition. At low resolution where unobserved high-frequency modes are significant, AMGFNO achieves up to 2–5$\times$ error reduction compared to S4FFNO. At high resolution, where most spectral information is observable, the gate reduces memory usage, matching or exceeding purely Markovian models. The gate adapts without manual tuning, with learned values adjusting according to resolution and spectral characteristics through training.

Future work includes extending AMGFNO to higher-dimensional PDEs, exploring alternative condition descriptors beyond spectral energy ratios, and evaluating robustness under observational noise and distribution shift.

Acknowlegments

This work was partly supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korean government (MSIT) (No. RS-2026-25526850, High-Efficiency Neural Networks for Artificial General Intelligence (HERMES-Net)), Samsung Electronics Co., Ltd. (No. G01240136, KAIST Semiconductor Research Fund (2nd)) and the Korea Advanced Institute of Science and Technology (KAIST) grant funded by the Korea government (MSIT) (No. G04240001, Physics-inspired Deep Learning). J. Choi was additionally supported by the KAIST Jang Young Sil Fellow Program.

References

Amann, H. Existence and regularity for semilinear parabolic evolution equations. *Annali della Scuola Normale Superiore di Pisa, Classe di Scienze*, 11(4):593–676, 1984.

Bach, F. *Learning Theory from First Principles*. Adaptive Computation and Machine Learning. The MIT Press, 2024. ISBN 9780262049443.

Brandstetter, J., Welling, M., and Worrall, D. E. Lie point symmetry data augmentation for neural pde solvers. In *International Conference on Machine Learning*, pp. 2241–2256. PMLR, 2022.

Buitrago Ruiz, R., Marwah, T., Gu, A., and Risteski, A. On the benefits of memory for modeling time-dependent PDEs. In *International Conference on Learning Representations*, 2025.

Cao, S. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34: 24924–24940, 2021.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

Cooley, J. W., Lewis, P. A. W., and Welch, P. D. The finite fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 17(2):77–85, 1969.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. HiPPO: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Gupta, J. K. and Brandstetter, J. Towards multi-spatiotemporal-scale generalized PDE modeling. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Hairer, E. and Wanner, G. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag, 1996.

Henry, D. *Geometric Theory of Semilinear Parabolic Equations*, volume 840 of *Lecture Notes in Mathematics*. Springer, 1981.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023a.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operator: Learning maps between function spaces. *Journal of Machine Learning Research*, 24(89):1–97, 2023b.

LeVeque, R. J. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

LeVeque, R. J. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Li, Z., Meidani, K., and Farimani, A. B. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023a.

Li, Z., Shu, D., and Farimani, A. B. Scalable transformer for PDE surrogate modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Lippe, P., Veeling, B., Perdikaris, P., Turner, R., and Brandstetter, J. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. *Advances in Neural Information Processing Systems*, 36:67398–67433, 2023.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Lu, L., Jin, P., and Karniadakis, G. E. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

Nicolaenko, B., Scheurer, B., and Temam, R. Some global dynamical properties of the kuramoto–sivashinsky equations: Nonlinear stability and attractors. *Physica D: Nonlinear Phenomena*, 16(2):155–183, 1985.

Parish, E. J. and Duraisamy, K. Non-markovian closure models for large eddy simulations using the mori-zwanzig formalism. *Physical Review Fluids*, 2(1):014604, 2017.

Pazy, A. *Semigroups of Linear Operators and Applications to Partial Differential Equations*, volume 44 of *Applied Mathematical Sciences*. Springer, 1983.

Robinson, J. C. *Infinite-Dimensional Dynamical Systems: An Introduction to Dissipative Parabolic PDEs and the Theory of Global Attractors*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2001.

Schiesser, W. E. *The Numerical Method of Lines*. Academic Press, 1991.

Shannon, C. E. Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, 37(1): 10–21, 1949.

Srivastava, R. K., Greff, K., and Schmidhuber, J. Training very deep networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

Tadmor, E. The well-posedness of the kuramoto–sivashinsky equation. *SIAM Journal on Mathematical Analysis*, 17(4):884–893, 1986.

Takamoto, M., Praditia, T., Leiteritz, R., MacKinlay, D., Alesiani, F., Pflüger, D., and Niepert, M. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Temam, R. *Infinite-Dimensional Dynamical Systems in Mechanics and Physics*, volume 68 of *Applied Mathematical Sciences*. Springer, 2 edition, 1997.

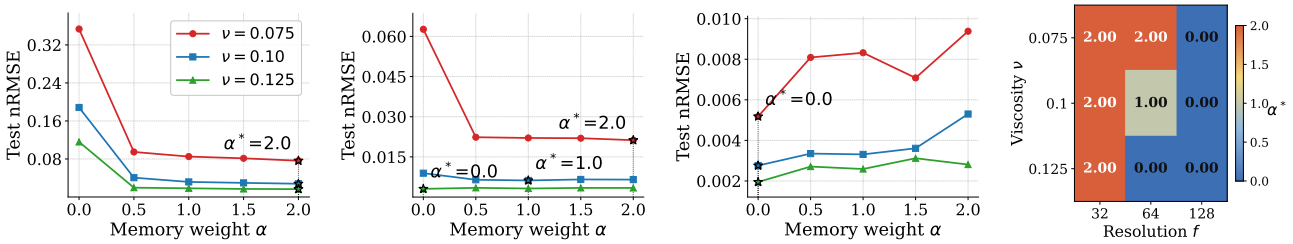
Tran, A., Mathews, A., Xie, L., and Ong, C. S. Factorized fourier neural operators. In *The Eleventh International Conference on Learning Representations*, 2023.

Tu, S., Frostig, R., and Soltanolkotabi, M. Learning from many trajectories. *Journal of Machine Learning Research*, 25(216):1–109, 2024.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

Zienkiewicz, O. C., Taylor, R. L., and Zhu, J. Z. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier Butterworth-Heinemann, 6 edition, 2005.

A. Preliminaries Study: Memory Ablation under Additive Fusion



(a) $f = 32$ *(b)* $f = 64$ *(c)* $f = 128$ *(d)* Optimal α^*

Figure 5. Effect of memory weight α on nRMSE across resolutions and viscosity values on the KS equation. Each panel shows nRMSE as a function of α for three viscosity values ($\nu \in \{0.075, 0.10, 0.125\}$). The rightmost heatmap summarizes optimal memory weight α^* per setting.

The motivation analysis in Section 2.2 parameterizes the memory weight via a convex blend, which is form-consistent with the adaptive memory gate of our main method. For completeness, we also report the same ablation under an additive parameterization that more directly mirrors the original residual fusion of S4FFNO (Buitrago Ruiz et al., 2025).

Using the same notation as in Section 2.2, let $\mathbf{h}_t \in \mathbb{R}^{N\times H}$ be the hidden state at the output of the second FFNO layer and $\mathbf{z}_t = \mathcal{M}(\mathbf{h}_{1:t}^{(\ell^*-1)})$ the output of the S4 memory branch. We replace the convex blend with the additive update

$$\mathbf{h}_{t+1} \leftarrow \alpha \cdot \mathbf{z}_t + \mathbf{h}_t, \tag{24}$$

which preserves the residual path in full and scales the memory weight by α. Under this parameterization, S4FFNO is recovered exactly at $\alpha = 1$, and $\alpha = 0$ again corresponds to a pure Markovian model that bypasses the memory branch. Allowing $\alpha > 1$ amplifies memory beyond its default contribution.

We sweep $\alpha \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$ on the Kuramoto–Sivashinsky equation under the same setting as Section 2.2 ($f \in \{32, 64, 128\}$, $\nu \in \{0.075, 0.1, 0.125\}$).

Observation 1: More memory is beneficial at low resolution. At $f = 32$, $\alpha^* = 2.0$ across all three viscosity values, and larger α consistently reduces nRMSE.

Observation 2: Memory isn't useful at high resolution. At $f = 128$, $\alpha^* = 0.0$ for all viscosity values, and larger α monotonically degrades performance.

Observation 3: The optimal memory weight α^* depends jointly on resolution and viscosity. At $f = 64$, α^* varies across viscosity values: $\alpha^* = 2.0$ at $\nu = 0.075$, $\alpha^* = 1.0$ at $\nu = 0.1$, and $\alpha^* = 0.0$ at $\nu = 0.125$. Higher viscosity reduces the high-frequency content of the PDE solution, diminishing the practical benefit of memory.

Implications. The same three qualitative conclusions emerge under both the convex (Section 2.2) and additive parameterizations: memory helps most at low resolution, hurts at high resolution, and its optimal weight depends jointly on resolution and viscosity. The robustness of these observations to the choice of fusion form supports our main motivation that no single fixed memory weight is appropriate across operating conditions, and that the gate must be learned and conditioned on the high-frequency structure of the input.

B. Choice of ω_f Computation: Fixed vs. Per-Batch

We compare two strategies for obtaining ω_f: (i) the proposed *fixed* variant, which estimates ω_f once as the training-set mean and holds it constant; and (ii) a *per-batch* variant that recomputes ω_f from each training and inference batch on the fly. Both variants share the same gate architecture, initialization, and training schedule. Table 4 reports final test nRMSE on the KS equation across all nine (f, ν) settings. The fixed variant consistently outperforms per-batch recomputation, with performance degradation ranging from 5.8% to 20.1% when ω_f is recomputed per batch. This supports our design choice in Section 3.3 to use a fixed, training-set-averaged ω_f rather than dynamic per-batch estimation.

Table 4. Final test nRMSE on KS for fixed vs. per-batch ω_f. The fixed variant outperforms per-batch recomputation in all 9 (f, ν) settings, with degradation ranging from 5.8% to 20.1%, supporting our design choice in Section 3.3.

f	ν	fixed	per-batch	$\Delta\%$
32	0.075	0.0616	0.0651	+5.8
32	0.10	0.0226	0.0249	+10.1
32	0.125	0.0142	0.0168	+18.2
64	0.075	0.0208	0.0250	+20.1
64	0.10	0.0069	0.0077	+11.9
64	0.125	0.0033	0.0039	+17.2
128	0.075	0.0058	0.0068	+18.5
128	0.10	0.0030	0.0033	+11.1
128	0.125	0.0019	0.0020	+5.8

Table 5. Ablation of the frequency-aware gate in AMGFNO. Bold indicates best performance per setting.

	Kuramoto–Sivashinsky									Burgers		
	$f = 32$			$f = 64$			$f = 128$					
Method	ν=.075	ν=.10	ν=.125	ν=.075	ν=.10	ν=.125	ν=.075	ν=.10	ν=.125	f=32	f=64	f=128
AMGFNO	0.0616	**0.0226**	**0.0142**	0.0208	0.0069	0.0033	**0.0058**	**0.0030**	**0.0019**	**0.0244**	0.0179	0.0202
w/o frequency-aware	**0.0573**	0.0231	0.0151	**0.0185**	**0.0058**	**0.0033**	0.0066	0.0041	0.0027	0.0253	**0.0140**	**0.0122**

We hypothesize that per-batch ω_f introduces noisy, sample-dependent signals that destabilize gate learning. In contrast, the fixed variant provides a stable global measure of spectral complexity tied to the resolution–viscosity condition, allowing the gate to learn consistent memory modulation patterns. The fixed approach also reduces computational overhead by eliminating redundant FFT operations at each forward pass.

C. Ablation: Effect of the Frequency-Aware Gate

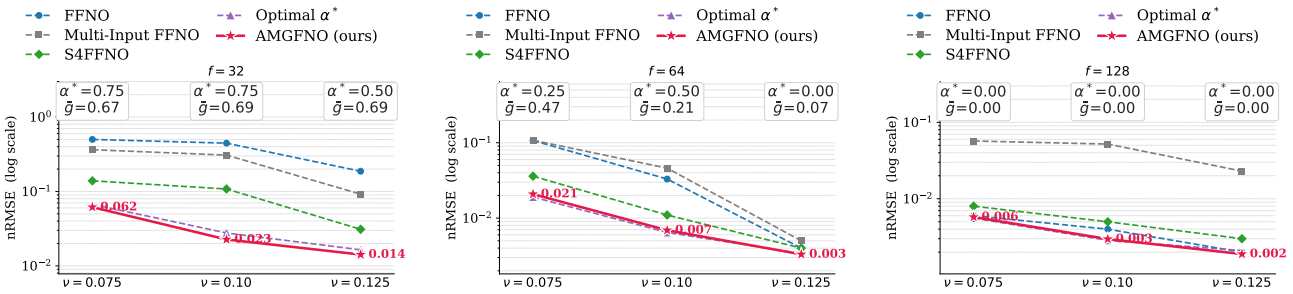
Table 5 reports the effect of removing the frequency-aware gate from AMGFNO, retaining only the content gate $\mathbf{g}_t^{(c)}$. The frequency-aware gate $\sigma(\omega_f \cdot w_1 + b_\omega)$ provides a spectral prior that modulates the baseline gate budget based on the fraction of unobserved high-frequency modes. We evaluate whether this prior offers a consistent advantage or whether the content gate alone can learn appropriate memory modulation from data.

KS: Frequency-aware gate helps at high resolution. On KS, ω_f varies dramatically across resolutions: $\omega_{32} = 0.584$, $\omega_{64} = 0.083$, $\omega_{128} = 0.0003$. At high resolution ($f = 128$), AMGFNO consistently outperforms w/o frequency-aware across all three viscosity settings. This aligns with Theorem 2.2: when $\omega_f \to 0$ (minimal high-frequency loss), the frequency-aware gate correctly drives the gate budget toward zero, suppressing unnecessary memory usage and avoiding the complexity cost of maintaining memory when it provides no benefit.

At low and mid resolution ($f \in \{32, 64\}$), results are mixed, with w/o frequency-aware winning in 4 out of 6 settings. However, AMGFNO still outperforms at the most challenging low-viscosity setting ($\nu = 0.075$, $f = 32$), where unobserved high-frequency modes are most prevalent and memory is essential.

Burgers: Frequency-aware gate has limited impact. On Burgers', ω_f is uniformly small across all resolutions: $\omega_{32} = 0.0062$, $\omega_{64} = 0.0012$, $\omega_{128} = 0.0002$. Here, w/o frequency-aware matches or outperforms AMGFNO at mid and high resolutions ($f \in \{64, 128\}$). When ω_f provides little differentiation across conditions, the frequency-aware gate offers minimal additional signal beyond what the content gate learns from data.

Implications. The frequency-aware gate is most valuable when ω_f varies meaningfully across observation conditions, as demonstrated at high resolution on KS where it consistently improves performance. When ω_f is uniformly small (as in Burgers'), the content gate alone suffices. This suggests a direction for future work: PDE-adaptive initialization of w_1 and b_ω based on the dataset's ω_f distribution could improve robustness across diverse PDE families. For instance, setting $w_1 \propto \max_f \omega_f - \min_f \omega_f$ would amplify the frequency-aware gate's influence when spectral variation is high and suppress it when variation is negligible.



(a) KS $f = 32$ *(b)* KS $f = 64$ *(c)* KS $f = 128$

Figure 6. Performance comparison on KS across resolutions $f \in \{32, 64, 128\}$ and viscosities $\nu \in \{0.075, 0.10, 0.125\}$. Optimal α^* denotes a model trained with the manually tuned memory weight α^* that achieves best performance per setting.

D. Performance Comparison Across Resolutions

Figure 6 extends Section 4.3 across all three resolutions. Optimal α^* represents an oracle baseline where the memory weight is manually tuned for each (f, ν) condition, providing an upper bound on what can be achieved with a fixed memory weight. AMGFNO achieves performance comparable to Optimal α^* across all conditions by learning adaptive memory gate values through training. At $f = 32$, where substantial high-frequency content is unobserved, AMGFNO learns gate values $\bar{g} = 0.67$–0.69 and achieves nRMSE of 0.062, 0.023, and 0.014 across the three viscosities, closely matching the oracle performance. At $f = 64$, the learned gate values adapt across viscosities ($\bar{g} = 0.47, 0.21, 0.07$), achieving nRMSE of 0.021, 0.007, and 0.003. At $f = 128$, where minimal high-frequency information is lost, AMGFNO learns $\bar{g} \approx 0.00$ and achieves nRMSE of 0.006, 0.003, and 0.002.

These results demonstrate that AMGFNO automatically learns condition-appropriate memory usage without requiring the exhaustive hyperparameter search needed to obtain Optimal α^*. The learned gate behavior is consistent with the Mori-Zwanzig principle: opening when unobserved high-frequency modes are prevalent and closing when the Markovian assumption suffices.

E. Relationship Between Gate Value and Training Loss

Figure 7 shows the co-evolution of test loss and adaptive memory gate value $\bar{g}$ over 200 training epochs. Test loss decreases monotonically while the gate converges to a stable level, with gate convergence typically preceding final loss convergence. The converged gate values reflect the need for memory. For KS, $\bar{g}$ stratifies by resolution and viscosity: at $f = 32$, $\bar{g} \approx 0.67$–0.70 across all viscosities; at $f = 64$, $\bar{g}$ ranges from 0.47 (low viscosity) to 0.07 (high viscosity); at $f = 128$, $\bar{g} \approx 0$ for all conditions. For Burgers', the gate converges to small values ($\bar{g} \approx 0.027$ at $f = 32$, decreasing to ≈ 0.001 at $f = 128$), consistent with uniformly small ω_f values. These dynamics confirm that the adaptive memory gate discovers appropriate memory usage purely through training.

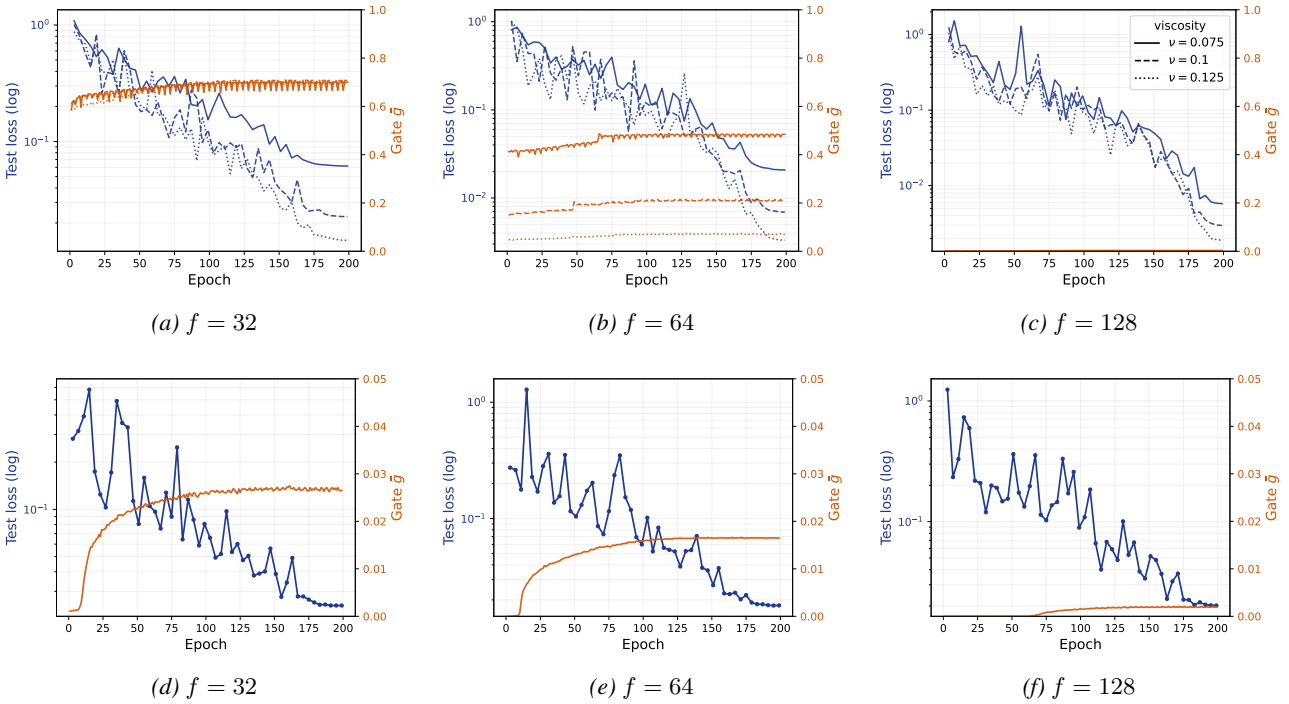
F. Experimental Details

This appendix provides the experimental details omitted from Section 4.1, including PDE equations, dataset construction, resolution processing, baseline architectures, training hyperparameters, evaluation protocol, and the computation of the spectral ratio used by AMGFNO. Unless otherwise stated, our experimental setup follows S4FFNO (Buitrago Ruiz et al., 2025). In particular, the Kuramoto–Sivashinsky datasets follow the low-viscosity benchmark generated in S4FFNO, which is based on the PDE-Refiner/LPSDA generation pipeline (Lippe et al., 2023; Brandstetter et al., 2022). The Burgers' dataset is taken from the public PDEBench benchmark (Takamoto et al., 2022).

F.1. PDE Datasets and Parameters

Kuramoto–Sivashinsky equation. The Kuramoto–Sivashinsky equation is given by

$$u_t(t,x) + u(t,x)u_x(t,x) + u_{xx}(t,x) + \nu u_{xxxx}(t,x) = 0, \qquad (t,x) \in [0,T] \times [0,L], \tag{25}$$



(a) $f = 32$ *(b)* $f = 64$ *(c)* $f = 128$

(d) $f = 32$ *(e)* $f = 64$ *(f)* $f = 128$

Figure 7. Evolution of test loss (blue, left axis, log scale) and adaptive memory gate value $\bar{g}$ (orange, right axis) over training epochs for AMGFNO. *(a, b, c)* KS equation at resolutions $f \in \{32, 64, 128\}$ for viscosities $\nu \in \{0.075, 0.10, 0.125\}$ (solid, dashed, and dotted lines, respectively). *(d, e, f)* Burgers' equation at resolutions $f \in \{32, 64, 128\}$ with $\nu = 0.001$.

with periodic boundary conditions. Following S4FFNO, we use low-viscosity KS datasets because low viscosity produces solutions with substantial high-frequency Fourier components. We use

$$\nu \in \{0.075, 0.10, 0.125\}. \tag{26}$$

The spatial domain length is

$$L = 64, \tag{27}$$

and the final rollout time used in our experiments is

$$T = 2.5. \tag{28}$$

Each trajectory is observed at 26 equispaced time points, giving 25 prediction timesteps with

$$\Delta t = 0.1. \tag{29}$$

The high-resolution reference trajectories are generated at spatial resolution 512. We use 2048 training trajectories and 256 held-out evaluation trajectories, following the S4FFNO setup.

The initial condition is a superposition of sinusoidal waves:

$$u_0(x) = \sum_{i=0}^{20} A_i \sin\left(\frac{2\pi k_i}{L} x + \phi_i\right), \tag{30}$$

where, for each trajectory,

$$A_i \sim \mathrm{Unif}[-0.5, 0.5], \qquad k_i \sim \mathrm{Unif}\{1, \ldots, 8\}, \qquad \phi_i \sim \mathrm{Unif}[0, 2\pi]. \tag{31}$$

The data generation follows the method of lines (Schiesser, 1991). Spatial derivatives are computed pseudospectrally using Fast Fourier transforms (Cooley et al., 1969), and the resulting ODE system is solved with an implicit Runge–Kutta method of the Radau IIA family of order 5 (Hairer & Wanner, 1996), implemented through the SciPy numerical stack (Virtanen et al., 2020).

Table 6. Dataset and rollout summary.

Dataset	ν	high-res. grid	train	rollout steps
KS	$\{0.075, 0.10, 0.125\}$	512	2048	25
Burgers'	0.001	1024	2048	20

Burgers' equation. The viscous Burgers' equation is given by

$$u_t(t,x) + u(t,x)u_x(t,x) = \nu u_{xx}(t,x), \qquad (t,x) \in [0,T] \times [0,L], \tag{32}$$

with periodic boundary conditions. We use the public PDEBench Burgers' dataset (Takamoto et al., 2022) with viscosity

$$\nu = 0.001. \tag{33}$$

The highest available spatial resolution is 1024. Following S4FFNO, we use 2048 samples for training and reserve 10% of the full dataset for testing. We evaluate the first 20 prediction timesteps because after this period the diffusion term attenuates most high-frequency components and the solution changes slowly. The final time is

$$T = 1.4, \tag{34}$$

and the rollout horizon contains 20 prediction timesteps.

F.2. Resolution Construction

For both PDEs, we evaluate all models at

$$f \in \{32, 64, 128\}. \tag{35}$$

For a domain of length L, the equispaced grid at resolution f is

$$S_f := \left\{ x_r = \frac{rL}{f} : r = 0, \dots, f-1 \right\}. \tag{36}$$

The high-resolution reference trajectories are first generated or loaded, and lower-resolution observations are constructed from the highest-resolution trajectory. For one-dimensional datasets, S4FFNO constructs lower-resolution trajectories by cubic interpolation from the highest-resolution grid. We follow the same procedure. Thus, for each resolution f, every model is trained and tested on the corresponding grid S_f.

For trajectory i and timestep j, the resolved state is

$$u_j^{f,(i)} := \mathcal{I}_f \left[u^{(i)}(t_j, \cdot) \right] \in \mathbb{R}^{f \times V}, \tag{37}$$

where $\mathcal{I}_f$ denotes cubic interpolation from the highest-resolution reference grid to S_f, and V is the number of physical channels. All models use the same resolved states $u_j^{f,(i)}$ and the same grid encoding.

Relation to the theory. Appendix G analyzes an ideal Fourier projection P_f in order to isolate the unresolved spectral-tail mechanism. The experiments follow the S4FFNO data protocol and construct lower-resolution inputs by cubic interpolation for comparability with prior work. The empirical quantity ω_f is nevertheless computed from high-resolution training trajectories using Fourier coefficients, as described in Appendix F.3.

F.3. Computation of the Spectral Ratio

By the Nyquist–Shannon sampling theorem (Shannon, 1949), an observation at spatial resolution f can resolve Fourier modes only up to approximately $\lfloor f/2 \rfloor$. For a full reference state ϕ, define the unresolved-mode ratio

$$\omega_f(\phi) := \frac{\sum_{|n| > \lfloor f/2 \rfloor} |a_n(\phi)|^2}{\sum_{n \in \mathbb{Z}} |a_n(\phi)|^2}, \tag{38}$$

where $a_n(\phi)$ denotes the Fourier coefficient of ϕ. For vector-valued states, the squared magnitude is summed over physical channels.

In practice, the continuous Fourier coefficients are approximated using the discrete Fourier transform of the highest-resolution trajectory available in the dataset (Cooley et al., 1969). For KS, the highest resolution is 512; for Burgers', it is 1024. For each condition (f, ρ), we compute the empirical training-set estimate

$$\widehat{\omega}_{f,\rho} := \frac{1}{N_{\text{train}} T_{\text{obs}}} \sum_{i=1}^{N_{\text{train}}} \sum_{j=0}^{T_{\text{obs}}-1} \omega_f \left(u_j^{\text{high},(i)} \right), \tag{39}$$

where $u_j^{\text{high},(i)}$ is the high-resolution training reference state. This provides a condition-level estimate of how much spectral energy is lost at each observation resolution. It is computed only from training trajectories and is held fixed during evaluation; no high-resolution test information is used.

The scalar frequency-aware gate in AMGFNO uses

$$\sigma \left(w_1 \log(\widehat{\omega}_{f,\rho} + \varepsilon_{\text{num}}) + b_\omega \right), \tag{40}$$

where $\varepsilon_{\text{num}} > 0$ is a small numerical constant used only for finite-precision stability. The asymptotic theory in Appendix G uses the corresponding zero-tail convention.

Note on observability. The condition-level estimate $\widehat{\omega}_{f,\rho}$ is computed from high-resolution training references and held fixed during training and evaluation. Thus, it is a calibration constant for the observation condition (f, ρ) rather than a per-sample test-time feature, and no high-resolution test information is used. For deployment to new conditions without high-resolution training references, this scalar could be replaced by observable proxies, such as condition-level summaries of the resolved high-band spectrum, or by metadata-based gates conditioned on f and available physical parameters ρ. We leave the systematic study of such proxies to future work.

F.4. Compared Methods and Implementation Details

We compare AMGFNO against both Markovian and memory-augmented neural operators. The baseline implementations and hyperparameters follow S4FFNO whenever possible.

Shared input and output processing. All models use a spatial grid encoding. In one dimension, for a grid of resolution f on $[0, L]$, we use the normalized coordinate

$$g_r := \frac{x_r}{L}, \qquad x_r = \frac{rL}{f}. \tag{41}$$

The encoder maps the concatenation of the PDE state and grid coordinate to the hidden dimension, and the decoder maps the final hidden representation back to the PDE state.

FFNO. FFNO is based on the Factorized Fourier Neural Operator (Tran et al., 2023), which builds on the original Fourier Neural Operator (Li et al., 2021). The FFNO layer combines a spectral convolution path with a pointwise feed-forward path. The hidden dimension is 128 and the expanded hidden dimension in the FFNO MLP is $4 \cdot 128$. We use all resolvable Fourier modes by setting

$$k_{\max} = \left\lfloor \frac{f}{2} \right\rfloor. \tag{42}$$

S4FFNO. S4FFNO instantiates the Memory Neural Operator framework by combining FFNO layers with an S4 memory layer (Buitrago Ruiz et al., 2025; Gu et al., 2022). All layers except the memory layer are the same as FFNO. The S4 layer uses state dimension 64 and the diagonal S4D kernel. Following the S4FFNO implementation, the memory layer operates along the temporal history dimension and is applied independently at each spatial location. The layer configuration is FFNO−FFNO−S4−FFNO−FFNO.

Table 7. Training hyperparameters used in the one-dimensional experiments.

Hyperparameter	KS	Burgers'
Epochs	200	200
Initial learning rate	10^{-3}	10^{-3}
Scheduler	cosine annealing	cosine annealing
Batch size	32	64
Training samples	2048	2048
Rollout steps	25	20

Multi-Input FFNO. Multi-Input FFNO uses the last $K = 4$ timesteps of the solution as input to predict the next timestep, following the multi-step input setting of FNO and S4FFNO (Li et al., 2021; Buitrago Ruiz et al., 2025). The number of FFNO layers and hidden dimensions are the same as FFNO. The only architectural change is the input lifting layer, which receives a concatenation of four previous states.

Factformer. Factformer is a transformer-based neural operator baseline (Li et al., 2023b). We use the one-dimensional adaptation described in S4FFNO. The model uses four linear attention layers over the spatial sequence. The hidden dimension is 64, with four heads and hidden dimension 128 in each attention layer.

Galerkin Transformer. The Galerkin Transformer baseline follows Cao (2021). It uses four linear attention layers over the spatial sequence, concatenates positional information into the attention inputs, and uses two FNO layers as a spectral regressor. The hidden dimension is 32.

U-Net. The U-Net neural operator baseline follows the U-Net style architecture used for PDE surrogate modeling (Gupta & Brandstetter, 2023). It consists of four downsample convolution blocks, a middle convolution block, and four upsample convolution blocks with skip connections. The channel multipliers are $[1, 2, 2, 2]$, and no time embeddings are used.

AMGFNO. AMGFNO uses the same FFNO backbone and S4 memory branch as S4FFNO, but replaces the fixed memory weight with the adaptive memory gate defined in Section 3.3. The gate combines a content-dependent component with the condition-level spectral scale derived from $\widehat{\omega}_{f,\rho}$. We do not redefine the gate here to avoid duplicating Section 3.3; this appendix specifies the shared experimental protocol.

F.5. Training Details

All one-dimensional experiments are trained with teacher forcing, following FFNO and S4FFNO (Tran et al., 2023; Buitrago Ruiz et al., 2025). During training, the input for predicting timestep $j + 1$ is the ground-truth resolved state at timestep j, rather than the model's previous prediction. For model $\mathcal{G}_\theta$, the one-step prediction is

$$\widetilde{u}_{j+1}^{f,(i)} = \mathcal{G}_\theta \left(u_j^{f,(i)}, g_f \right), \tag{43}$$

where g_f denotes the spatial grid encoding. The mini-batch training loss is

$$\mathcal{L}_{\text{train}}(\theta) = \frac{1}{|\mathcal{B}| T_{\text{pred}}} \sum_{i \in \mathcal{B}} \sum_{j=0}^{T_{\text{pred}}-1} \left\| \mathcal{G}_\theta \left(u_j^{f,(i)}, g_f \right) - u_{j+1}^{f,(i)} \right\|_2^2 . \tag{44}$$

All models are trained for 200 epochs with initial learning rate

$$10^{-3}, \tag{45}$$

using cosine annealing learning-rate scheduling (Loshchilov & Hutter, 2017). The batch size is 32 for KS and 64 for Burgers'. Both KS and Burgers' experiments use 2048 training samples. The maximum memory length is the full trajectory length used in the rollout: 25 prediction timesteps for KS and 20 prediction timesteps for Burgers'.

We ran our experiments on NVIDIA A6000 GPUs. The one-dimensional experiments require less than 10GB of GPU memory and take approximately 1–2 hours per run, depending on the resolution, following the computational condition reported by S4FFNO.

F.6. Evaluation

At test time, all models are evaluated autoregressively from the first observed state. The rollout is initialized by

$$\widetilde{u}_0^{f,(i)} = u_0^{f,(i)}, \tag{46}$$

and predictions are fed back into the model:

$$\widetilde{u}_{j+1}^{f,(i)} = \mathcal{G}_\theta\left(\widetilde{u}_j^{f,(i)}, g_f\right), \qquad j = 0, \ldots, T_{\text{pred}} - 1. \tag{47}$$

Thus, the reported metric evaluates accumulated rollout error rather than one-step prediction error.

The main evaluation metric is normalized root mean squared error, nRMSE. For trajectory i and timestep j, define

$$\text{nRMSE}_j^{(i)}(f) = \frac{\left\|\widetilde{u}_j^{f,(i)} - u_j^{f,(i)}\right\|_2}{\left\|u_j^{f,(i)}\right\|_2}. \tag{48}$$

The final reported score is averaged over test trajectories and rollout timesteps:

$$\text{nRMSE}(f) = \frac{1}{N_{\text{test}} T_{\text{pred}}} \sum_{i=1}^{N_{\text{test}}} \sum_{j=1}^{T_{\text{pred}}} \text{nRMSE}_j^{(i)}(f). \tag{49}$$

The values in Table 2 are computed using this rollout nRMSE.

For AMGFNO, we additionally measure the mean gate value during rollout. For a test trajectory, let $g_j^{(i)} \in (0,1)^{f \times H}$ denote the adaptive memory gate at timestep j. We define

$$\overline{g}(f, \rho) = \frac{1}{N_{\text{test}} T_{\text{pred}} f H} \sum_{i=1}^{N_{\text{test}}} \sum_{j=0}^{T_{\text{pred}}-1} \left\|g_j^{(i)}\right\|_1. \tag{50}$$

This value quantifies how strongly AMGFNO uses the memory path under a given resolution and physical condition. The values reported in Table 3 are computed as the mean gate value over the last 20 epochs.

G. Detailed Theory and Proofs for AMGFNO

This appendix provides the formal details behind the theoretical motivation in Section 2.3. The main text keeps the high-level argument and the frequency-gate quantities ω_f, $\omega_{f,\rho}^{\text{pop}}$, $\widehat{\omega}_{f,\rho}$, $\chi_{f,\rho}$, and $\lambda_{f,\rho}$ explicit, while deferring the full prediction-target, hypothesis-class, approximation-error, and finite-sample definitions to this appendix.

The comparison treats FFNO, S4FFNO, and AMGFNO as functions of the same resolved history. FFNO is the restricted subclass that uses only the current resolved state, S4FFNO uses a fixed residual memory fusion, and AMGFNO uses the same S4 memory branch with an adaptive memory gate. This shared-history view separates two effects: the approximation benefit of memory and the finite-sample complexity cost of maintaining a memory branch.

The notation is aligned with the AMGFNO gate in Section 3.3. We use u_t for the PDE state, h_t for the Markovian hidden feature at the AMG insertion point, z_t for the S4 memory output, and g_t for the adaptive memory gate. The symbol $\mathcal{F}$ is reserved for the Fourier transform in FFNO layers, so predictor classes are denoted by $\mathcal{C}$ and loss classes by $\mathfrak{L}$. The architectural setting follows FFNO-style neural operators and S4-based temporal memory models (Gu et al., 2020; Tran et al., 2023; Gu et al., 2022; Buitrago Ruiz et al., 2025).

All architecture classes below are identified with their L^2 closures under the population distribution induced by the data-generating process. This convention lets us use limiting cases such as zero memory output and zero gate in approximation arguments, even though sigmoid gates take values in $(0,1)$ before closure.

Proof roadmap. Appendix G.4 proves Theorem 2.1. The proof first bounds the influence of the unresolved Fourier tail, then transfers approximation of the projected Markovian target to approximation of the history-conditioned Bayes target. Appendix G.8 proves Theorem 2.2. The proof combines spectral gate convergence, finite-sample Rademacher bounds, and a sufficient condition under which the AMGFNO complexity gap closes as its gate budget shrinks.

G.1. Setup and Common Bayes Target

We state the argument on the one-dimensional periodic domain $\mathbb{T}$. The same notation applies to $[0, L]$ after replacing $e^{in\xi}$ by $e^{i2\pi n\xi/L}$. Let

$$\mathcal{H} := L^2(\mathbb{T}; \mathbb{R}^V).$$

For a physical condition ρ, let

$$u^\rho_{t+1} = \Phi^\rho_{\Delta t}(u^\rho_t), \qquad u^\rho_0 \sim \mu_\rho, \tag{51}$$

where $\Phi^\rho_{\Delta t}$ is the time-Δt solution map. For $\phi \in \mathcal{H}$, write

$$\phi(\xi) = \sum_{n\in\mathbb{Z}} a_n(\phi) e^{in\xi}. \tag{52}$$

For vector-valued states, $a_n(\phi) \in \mathbb{C}^V$ and $|a_n(\phi)|^2$ denotes the squared Euclidean norm over channels.

Let $m_f := \lfloor f/2 \rfloor$. The ideal spectral observation at resolution f is

$$P_f\phi := \sum_{|n|\le m_f} a_n(\phi) e^{in\xi}, \qquad Q_f := I - P_f, \qquad \mathcal{V}_f := P_f\mathcal{H}. \tag{53}$$

Thus $Q_f\phi$ is the unresolved Fourier tail. The cutoff $|n| > m_f$ follows the Nyquist–Shannon interpretation of which Fourier modes are unavailable at resolution f (Shannon, 1949). The theory uses this idealized spectral projection to isolate the unresolved-tail mechanism.

The resolved state, resolved history, and one-step resolved target are

$$u^\rho_{t,f} := P_f u^\rho_t, \qquad \mathcal{U}^{f,\rho}_t := (u^\rho_{0,f}, \dots, u^\rho_{t,f}), \qquad Y^{f,\rho}_t := P_f u^\rho_{t+1}. \tag{54}$$

For clarity, the statements below are written for a fixed finite time index t. If t is sampled uniformly from a finite prediction horizon, all expectations are replaced by finite averages over t, and the arguments are unchanged.

For a measurable predictor $\psi : \mathcal{U}^{f,\rho}_t \mapsto \mathcal{V}_f$, define the population risk

$$R_{f,\rho}(\psi) := \mathbb{E}\left\| Y^{f,\rho}_t - \psi(\mathcal{U}^{f,\rho}_t) \right\|^2_{\mathcal{V}_f}. \tag{55}$$

The projected Markovian target is

$$\mathcal{T}_{f,\rho}(\phi) := P_f \Phi^\rho_{\Delta t}(\phi), \qquad \phi \in \mathcal{V}_f. \tag{56}$$

The resolved target decomposes as

$$Y^{f,\rho}_t = \mathcal{T}_{f,\rho}(u^\rho_{t,f}) + \delta_{t,f,\rho}, \tag{57}$$

where

$$\delta_{t,f,\rho} := P_f\Phi^\rho_{\Delta t}(u^\rho_t) - P_f\Phi^\rho_{\Delta t}(P_f u^\rho_t). \tag{58}$$

The residual $\delta_{t,f,\rho}$ is the effect of the unresolved tail $Q_f u^\rho_t$ on the next resolved state.

Proposition G.1 (Bayes decomposition)**.** *Define the history-conditioned Bayes target*

$$b_{f,\rho}(\mathcal{U}^{f,\rho}_t) := \mathbb{E}\left[Y^{f,\rho}_t \mid \mathcal{U}^{f,\rho}_t \right] \tag{59}$$

and the irreducible error

$$\varepsilon^2_{f,\rho} := \mathbb{E}\left\| Y^{f,\rho}_t - b_{f,\rho}(\mathcal{U}^{f,\rho}_t) \right\|^2_{\mathcal{V}_f}. \tag{60}$$

For every square-integrable predictor ψ,

$$R_{f,\rho}(\psi) = \varepsilon^2_{f,\rho} + \mathbb{E}\left\| b_{f,\rho}(\mathcal{U}^{f,\rho}_t) - \psi(\mathcal{U}^{f,\rho}_t) \right\|^2_{\mathcal{V}_f}. \tag{61}$$

Proof. Write $U = \mathcal{U}_t^{f,\rho}$, $Y = Y_t^{f,\rho}$, and $b(U) = \mathbb{E}[Y \mid U]$. Then

$$Y - \psi(U) = \big(Y - b(U)\big) + \big(b(U) - \psi(U)\big).$$

Expanding the squared norm gives

$$\begin{aligned}\mathbb{E}\|Y - \psi(U)\|_{\mathcal{V}_f}^2 = \mathbb{E}\|Y - b(U)\|_{\mathcal{V}_f}^2 + \mathbb{E}\|b(U) - \psi(U)\|_{\mathcal{V}_f}^2 \\ + 2\mathbb{E}\,\langle Y - b(U), b(U) - \psi(U)\rangle_{\mathcal{V}_f}\,.\end{aligned} \tag{62}$$

The cross term is zero because $b(U) - \psi(U)$ is $\sigma(U)$-measurable and

$$\mathbb{E}[Y - b(U) \mid U] = \mathbb{E}[Y \mid U] - b(U) = 0.$$

This proves the decomposition. □

G.2. Architecture Classes and Approximation Gains

Let

$$h_{k,\theta} := h_k^{(\ell^*-1)} \in \mathbb{R}^{S\times H}$$

be the hidden feature produced from $u_{k,f}^{\rho}$ by the encoder and the FFNO layers before the AMG insertion point. Let

$$z_{t,\theta} := M_\theta(h_{0,\theta}, \ldots, h_{t,\theta}) \in \mathbb{R}^{S\times H} \tag{63}$$

be the S4 memory output. The S4 branch applies a state-space recurrence along the temporal axis at each spatial position, treating the spatial dimension as a batch axis. Let Dec_θ denote the remaining FFNO layers and output decoder.

The FFNO class is

$$\mathcal{C}_{f,\rho}^{F} := \left\{\psi_\theta(\mathcal{U}_t^{f,\rho}) = \mathrm{Dec}_\theta(h_{t,\theta})\right\}. \tag{64}$$

This class is formally defined on the resolved history but ignores all states except the current one.

The fixed-fusion S4FFNO class is

$$\mathcal{C}_{f,\rho}^{S4} := \left\{\psi_\theta(\mathcal{U}_t^{f,\rho}) = \mathrm{Dec}_\theta(h_{t,\theta} + z_{t,\theta})\right\}. \tag{65}$$

For the content-gated ablation without the frequency-aware gate, define

$$\mathcal{C}_{f,\rho}^{CG} := \left\{\psi_\theta(\mathcal{U}_t^{f,\rho}) = \mathrm{Dec}_\theta\left((1 - g_{t,\theta}^{\mathrm{cont}}) \odot h_{t,\theta} + g_{t,\theta}^{\mathrm{cont}} \odot z_{t,\theta}\right)\right\}, \tag{66}$$

where

$$g_{t,\theta}^{\mathrm{cont}} := \sigma(W_z z_{t,\theta} + W_h h_{t,\theta} + b) \in (0,1)^{S\times H}. \tag{67}$$

For a full reference state $\phi \in \mathcal{H}$, define the unresolved spectral-energy ratio

$$\omega_f(\phi) := \frac{\sum_{|n|>m_f} |a_n(\phi)|^2}{\sum_{n\in\mathbb{Z}} |a_n(\phi)|^2}, \tag{68}$$

with value zero when the denominator is zero. For condition (f,ρ), define

$$\omega_{f,\rho}^{\mathrm{pop}} := \mathbb{E}_{u_0,t}\left[\omega_f(u_t^{\rho})\right]. \tag{69}$$

In implementation, $\omega_{f,\rho}^{\mathrm{pop}}$ is replaced by the empirical training-set estimate $\widehat{\omega}_{f,\rho}$ computed from high-resolution training trajectories, with Fourier coefficients approximated by the discrete Fourier transform (Cooley et al., 1969). The notation ω_f in Section 3.3 denotes this condition-level scalar when (f,ρ) is fixed. It is not computed from unresolved test-time modes.

Define the scalar frequency gate by

$$\chi_{f,\rho} := \begin{cases}\sigma\Big(w_1 \log \omega_{f,\rho}^{\mathrm{pop}} + b_\omega\Big), & \omega_{f,\rho}^{\mathrm{pop}} > 0,\\ 0, & \omega_{f,\rho}^{\mathrm{pop}} = 0,\end{cases} \qquad w_1 > 0. \tag{70}$$

The numerical implementation evaluates $\log(\widehat{\omega}_{f,\rho} + \varepsilon_{\mathrm{num}})$ for finite-precision stability. The asymptotic analysis uses the zero-tail convention in (70).

The AMGFNO gate and predictor are

$$g_{t,\theta} := \chi_{f,\rho} g_{t,\theta}^{\mathrm{cont}}, \tag{71}$$

and

$$\psi_\theta(\mathcal{U}_t^{f,\rho}) = \mathrm{Dec}_\theta\left((1 - g_{t,\theta}) \odot h_{t,\theta} + g_{t,\theta} \odot z_{t,\theta}\right). \tag{72}$$

The corresponding AMGFNO class is denoted by $\mathcal{C}_{f,\rho}^{AMG}$.

For $\psi \in \mathcal{C}_{f,\rho}^{AMG}$, define the normalized gate budget

$$\Lambda_{f,\rho}(\psi) := \mathbb{E}\left[\frac{1}{SH}\|g_{t,\theta}\|_F^2\right]. \tag{73}$$

For $\lambda \geq 0$, define

$$\mathcal{C}_{f,\rho}^{AMG}(\lambda) := \left\{\psi \in \mathcal{C}_{f,\rho}^{AMG} : \Lambda_{f,\rho}(\psi) \leq \lambda\right\}. \tag{74}$$

Since $0 \leq g_{t,\theta}^{\mathrm{cont}} \leq 1$ entrywise,

$$\Lambda_{f,\rho}(\psi) \leq \chi_{f,\rho}^2. \tag{75}$$

We write

$$\lambda_{f,\rho} := \chi_{f,\rho}^2. \tag{76}$$

Proposition G.2 (Markovian fallback)**.** *Assume the S4 parameterization contains a zero-output realization, for example by setting its readout to zero. Under the L^2-closure convention,*

$$\mathcal{C}_{f,\rho}^{F} \subset \mathcal{C}_{f,\rho}^{S4}, \qquad \mathcal{C}_{f,\rho}^{F} \subset \mathcal{C}_{f,\rho}^{CG}, \qquad \mathcal{C}_{f,\rho}^{F} \subset \mathcal{C}_{f,\rho}^{AMG}(\lambda) \tag{77}$$

for every $\lambda \geq 0$.

Proof. For S4FFNO, choose the S4 readout so that $z_{t,\theta} = 0$ for every history. Then

$$\mathrm{Dec}_\theta(h_{t,\theta} + z_{t,\theta}) = \mathrm{Dec}_\theta(h_{t,\theta}),$$

which is an FFNO predictor.

For the gated classes, take a sequence of gate biases tending to $-\infty$, so that the gate converges to zero in L^2. The fused feature converges to

$$(1 - g_{t,\theta}) \odot h_{t,\theta} + g_{t,\theta} \odot z_{t,\theta} \to h_{t,\theta}.$$

The resulting predictor is $\mathrm{Dec}_\theta(h_{t,\theta})$, and the gate budget is zero. Hence the FFNO predictor belongs to the gated classes in closure. □

Define the approximation errors to the common Bayes target:

$$\mathcal{E}_{f,\rho}^{F} := \inf_{\psi \in \mathcal{C}_{f,\rho}^{F}} \mathbb{E}\left\|b_{f,\rho}(\mathcal{U}_t^{f,\rho}) - \psi(\mathcal{U}_t^{f,\rho})\right\|_{\mathcal{V}_f}^2, \tag{78}$$

$$\mathcal{E}_{f,\rho}^{S4} := \inf_{\psi \in \mathcal{C}_{f,\rho}^{S4}} \mathbb{E}\left\|b_{f,\rho}(\mathcal{U}_t^{f,\rho}) - \psi(\mathcal{U}_t^{f,\rho})\right\|_{\mathcal{V}_f}^2, \tag{79}$$

$$\mathcal{E}_{f,\rho}^{CG} := \inf_{\psi \in \mathcal{C}_{f,\rho}^{CG}} \mathbb{E}\left\|b_{f,\rho}(\mathcal{U}_t^{f,\rho}) - \psi(\mathcal{U}_t^{f,\rho})\right\|_{\mathcal{V}_f}^2, \tag{80}$$

$$\mathcal{E}_{f,\rho}^{AMG}(\lambda) := \inf_{\psi \in \mathcal{C}_{f,\rho}^{AMG}(\lambda)} \mathbb{E}\left\|b_{f,\rho}(\mathcal{U}_t^{f,\rho}) - \psi(\mathcal{U}_t^{f,\rho})\right\|_{\mathcal{V}_f}^2. \tag{81}$$

The approximation gains over FFNO are

$$\Gamma^{S4}_{f,\rho} := \mathcal{E}^{F}_{f,\rho} - \mathcal{E}^{S4}_{f,\rho}, \tag{82}$$

$$\Gamma^{CG}_{f,\rho} := \mathcal{E}^{F}_{f,\rho} - \mathcal{E}^{CG}_{f,\rho}, \tag{83}$$

$$\Gamma^{AMG}_{f,\rho}(\lambda) := \mathcal{E}^{F}_{f,\rho} - \mathcal{E}^{AMG}_{f,\rho}(\lambda). \tag{84}$$

By Proposition G.2, these gains are nonnegative.

Finally, define the FFNO approximation error to the projected Markovian target:

$$\mathcal{A}^{F}_{f,\rho} := \inf_{\psi \in \mathcal{C}^{F}_{f,\rho}} \mathbb{E} \left\| \mathcal{T}_{f,\rho}(u^{\rho}_{t,f}) - \psi(\mathcal{U}^{f,\rho}_{t}) \right\|^2_{\mathcal{V}_f}. \tag{85}$$

Why distinguish $\mathcal{A}^{F}_{f,\rho}$ and $\mathcal{E}^{F}_{f,\rho}$? The distinction is central to the argument. $\mathcal{A}^{F}_{f,\rho}$ measures how well FFNO can approximate the ideal projected Markovian target $\mathcal{T}_{f,\rho}(u^{\rho}_{t,f})$, which depends only on the current resolved state. This is the approximation quantity appearing in Assumption G.5. In contrast, $\mathcal{E}^{F}_{f,\rho}$ measures FFNO's approximation error to the history-conditioned Bayes target $b_{f,\rho}(\mathcal{U}^{f,\rho}_{t})$, which is the actual partially observed prediction target and may depend on the entire resolved history. Therefore, the theory does not assume that FFNO already approximates the history-conditioned Bayes target. Instead, Lemma G.9 uses the unresolved-tail residual bound to transfer approximation of the projected Markovian target into a bound on the Bayes-target error.

G.3. Assumptions

Assumption G.3 (Finite-horizon well-posedness and Sobolev bound)**.** For μ_ρ-almost every initial condition, the PDE admits a unique solution over the finite prediction horizon. For some $r \geq 0$ and $\beta > 0$, there exists $R_{\rho,\beta} < \infty$ such that

$$\sup_{0 \leq t \leq T_{\text{pred}}} \|u^{\rho}_{t}\|_{H^{r+\beta}(\mathbb{T})} \leq R_{\rho,\beta} \qquad \text{almost surely.} \tag{86}$$

For dissipative long-time settings, this may be replaced by an absorbing-ball bound in $H^{r+\beta}$. Such well-posedness, finite-horizon regularity, and dissipativity assumptions are standard for semilinear parabolic and dissipative PDEs (Henry, 1981; Pazy, 1983; Amann, 1984; Temam, 1997; Robinson, 2001). For the Kuramoto–Sivashinsky equation, they are supported by the KS well-posedness and attractor literature (Nicolaenko et al., 1985; Tadmor, 1986).

Assumption G.4 (Local Lipschitz continuity of the solution map)**.** For every $R > 0$, there exists $L_{\rho,R} < \infty$ such that

$$\|\Phi^{\rho}_{\Delta t}(\phi) - \Phi^{\rho}_{\Delta t}(\psi)\|_{L^2(\mathbb{T})} \leq L_{\rho,R} \|\phi - \psi\|_{H^r(\mathbb{T})} \tag{87}$$

whenever $\|\phi\|_{H^r} \leq R$ and $\|\psi\|_{H^r} \leq R$. This is the finite-time continuous-dependence property used in semigroup treatments of semilinear parabolic equations (Henry, 1981; Pazy, 1983; Amann, 1984).

Assumption G.5 (FFNO approximation of the projected Markovian target)**.** The projected-target approximation error satisfies

$$\mathcal{A}^{F}_{f,\rho} \to 0 \qquad \text{as } f \to \infty. \tag{88}$$

This assumption concerns only the ideal projected Markovian target $\mathcal{T}_{f,\rho}$, not the history-conditioned Bayes target $b_{f,\rho}$. Thus it is not an assumption that FFNO can already solve the partially observed prediction problem. Rather, it states that, as the resolved space becomes richer, the FFNO class can approximate the Markovian solution map restricted to the resolved variables. This is consistent with neural-operator approximation results for continuous solution operators on compact sets (Li et al., 2021; Tran et al., 2023; Kovachki et al., 2023b).

Assumption G.6 (Bounded or sub-Gaussian losses)**.** For the finite-sample bounds, assume squared losses are uniformly bounded by ℓ_∞. Equivalently, one may replace this by a standard sub-Gaussian condition and obtain corresponding high-probability concentration terms. The bounded-loss form is used only to keep the comparison concise (Bach, 2024).

Assumption G.7 (Trajectory-level sampling)**.** Training trajectories are independent. Multiple timesteps from the same trajectory may be dependent. The finite-sample statements treat each trajectory as one independent sample and define the empirical loss as the average over the finite prediction horizon. This follows the standard multi-trajectory learning setup, where trajectories are independent but covariates within a trajectory need not be independent (Tu et al., 2024).

G.4. Residual Decay and Proof of Theorem 2.1

Lemma G.8 (Residual decay)**.** *Under Assumptions G.3 and G.4, there exists a constant $\kappa_{\mathrm{tail},\rho} > 0$, independent of f, such that*

$$\mathbb{E}\|\delta_{t,f,\rho}\|^2_{\mathcal{V}_f} \le \kappa_{\mathrm{tail},\rho} f^{-2\beta}. \tag{89}$$

Proof. Since P_f is an orthogonal projection in L^2,

$$\begin{aligned}\|\delta_{t,f,\rho}\|_{\mathcal{V}_f} &= \|P_f\left(\Phi^\rho_{\Delta t}(u^\rho_t) - \Phi^\rho_{\Delta t}(P_f u^\rho_t)\right)\|_{L^2} \\ &\le \|\Phi^\rho_{\Delta t}(u^\rho_t) - \Phi^\rho_{\Delta t}(P_f u^\rho_t)\|_{L^2}\,.\end{aligned} \tag{90}$$

By Assumption G.3, $\|u^\rho_t\|_{H^{r+\beta}} \le R_{\rho,\beta}$ almost surely. Since P_f is nonexpansive in Sobolev norms,

$$\|P_f u^\rho_t\|_{H^r} \le \|u^\rho_t\|_{H^r} \le \|u^\rho_t\|_{H^{r+\beta}} \le R_{\rho,\beta}.$$

Thus both u^ρ_t and $P_f u^\rho_t$ lie in the same H^r ball. Applying Assumption G.4 gives

$$\|\delta_{t,f,\rho}\|_{\mathcal{V}_f} \le L_{\rho,R_{\rho,\beta}}\|u^\rho_t - P_f u^\rho_t\|_{H^r} = L_{\rho,R_{\rho,\beta}}\|Q_f u^\rho_t\|_{H^r}. \tag{91}$$

Using the Fourier definition of the H^r norm,

$$\begin{aligned}\|Q_f u^\rho_t\|^2_{H^r} &= \sum_{|n|>m_f} (1+|n|^2)^r |a_n(u^\rho_t)|^2 \\ &= \sum_{|n|>m_f} (1+|n|^2)^{r+\beta}(1+|n|^2)^{-\beta}|a_n(u^\rho_t)|^2.\end{aligned} \tag{92}$$

For $|n| > m_f$, there exists C_β independent of f such that

$$(1+|n|^2)^{-\beta} \le (1+m_f^2)^{-\beta} \le C_\beta f^{-2\beta}.$$

Therefore

$$\|Q_f u^\rho_t\|^2_{H^r} \le C_\beta f^{-2\beta}\|u^\rho_t\|^2_{H^{r+\beta}} \le C_\beta f^{-2\beta} R^2_{\rho,\beta}. \tag{93}$$

Combining this with (91) and taking expectations proves the claim with

$$\kappa_{\mathrm{tail},\rho} := C_\beta L^2_{\rho,R_{\rho,\beta}} R^2_{\rho,\beta}.$$

□

Lemma G.9 (From projected-target approximation to Bayes-target approximation)**.** *Under Assumptions G.3 and G.4,*

$$\mathcal{E}^F_{f,\rho} \le 2\mathcal{A}^F_{f,\rho} + 2\kappa_{\mathrm{tail},\rho} f^{-2\beta}. \tag{94}$$

If Assumption G.5 also holds, then $\mathcal{E}^F_{f,\rho} \to 0$ as $f \to \infty$.

Proof. The proof transfers approximation of the projected Markovian target to approximation of the true history-conditioned Bayes target. From (57),

$$Y^{f,\rho}_t = \mathcal{T}_{f,\rho}(u^\rho_{t,f}) + \delta_{t,f,\rho}.$$

Taking conditional expectation with respect to $\mathcal{U}^{f,\rho}_t$ gives

$$b_{f,\rho}(\mathcal{U}^{f,\rho}_t) = \mathcal{T}_{f,\rho}(u^\rho_{t,f}) + c_{t,f,\rho}(\mathcal{U}^{f,\rho}_t), \tag{95}$$

where

$$c_{t,f,\rho}(\mathcal{U}^{f,\rho}_t) := \mathbb{E}\left[\delta_{t,f,\rho} \mid \mathcal{U}^{f,\rho}_t\right].$$

The term $\mathcal{T}_{f,\rho}(u^\rho_{t,f})$ is $\sigma(\mathcal{U}^{f,\rho}_t)$-measurable.

For any $\psi \in \mathcal{C}^F_{f,\rho}$,

$$\mathbb{E}\left\| b_{f,\rho}(\mathcal{U}_t^{f,\rho}) - \psi(\mathcal{U}_t^{f,\rho}) \right\|_{\mathcal{V}_f}^2 \leq 2\mathbb{E}\left\| \mathcal{T}_{f,\rho}(u^\rho_{t,f}) - \psi(\mathcal{U}_t^{f,\rho}) \right\|_{\mathcal{V}_f}^2 + 2\mathbb{E}\left\| c_{t,f,\rho}(\mathcal{U}_t^{f,\rho}) \right\|_{\mathcal{V}_f}^2 . \tag{96}$$

By Jensen's inequality for conditional expectation,

$$\mathbb{E}\left\| c_{t,f,\rho}(\mathcal{U}_t^{f,\rho}) \right\|_{\mathcal{V}_f}^2 = \mathbb{E}\left\| \mathbb{E}[\delta_{t,f,\rho} \mid \mathcal{U}_t^{f,\rho}] \right\|_{\mathcal{V}_f}^2 \leq \mathbb{E}\|\delta_{t,f,\rho}\|^2_{\mathcal{V}_f}. \tag{97}$$

Taking the infimum over $\psi \in \mathcal{C}^F_{f,\rho}$ in (96) and applying Lemma G.8 proves (94). If $\mathcal{A}^F_{f,\rho} \to 0$, the right-hand side converges to zero. □

Proof of Theorem 2.1. Lemma G.9 gives

$$\mathcal{E}^F_{f,\rho} \leq 2\mathcal{A}^F_{f,\rho} + 2\kappa_{\mathrm{tail},\rho} f^{-2\beta}. \tag{98}$$

Under Assumption G.5, $\mathcal{A}^F_{f,\rho} \to 0$. Since $f^{-2\beta} \to 0$, this implies

$$\mathcal{E}^F_{f,\rho} \to 0.$$

Importantly, the convergence of $\mathcal{E}^F_{f,\rho}$ is a conclusion of the residual-transfer bound, not an assumption.

By Proposition G.2, each memory-augmented class contains the FFNO fallback in closure. Hence

$$\mathcal{E}^{S4}_{f,\rho} \leq \mathcal{E}^F_{f,\rho}, \qquad \mathcal{E}^{AMG}_{f,\rho}(\lambda) \leq \mathcal{E}^F_{f,\rho}$$

for every $\lambda \geq 0$. Therefore the corresponding memory gains satisfy

$$0 \leq \Gamma^{S4}_{f,\rho} = \mathcal{E}^F_{f,\rho} - \mathcal{E}^{S4}_{f,\rho} \leq \mathcal{E}^F_{f,\rho},$$

and

$$0 \leq \Gamma^{AMG}_{f,\rho}(\lambda) = \mathcal{E}^F_{f,\rho} - \mathcal{E}^{AMG}_{f,\rho}(\lambda) \leq \mathcal{E}^F_{f,\rho}.$$

Since $\mathcal{E}^F_{f,\rho} \to 0$, both gains converge to zero by squeezing. This proves Theorem 2.1. □

G.5. Spectral Gate Convergence

Lemma G.10 (Condition-level spectral gate convergence)**.** *Let $\omega_f(\phi)$, $\omega^{\mathrm{pop}}_{f,\rho}$, $\chi_{f,\rho}$, and $\lambda_{f,\rho}$ be defined by (68)–(76). Then*

$$\omega^{\mathrm{pop}}_{f,\rho} \to 0, \qquad \chi_{f,\rho} \to 0, \qquad \lambda_{f,\rho} = \chi^2_{f,\rho} \to 0 \qquad \text{as } f \to \infty. \tag{99}$$

Moreover, every $\psi \in \mathcal{C}^{AMG}_{f,\rho}$ satisfies

$$\Lambda_{f,\rho}(\psi) \leq \lambda_{f,\rho}. \tag{100}$$

Proof. Fix any $u^\rho_t \in L^2(\mathbb{T};\mathbb{R}^V)$. Its Fourier coefficients are square summable:

$$\sum_{n\in\mathbb{Z}} |a_n(u^\rho_t)|^2 < \infty.$$

Therefore

$$\sum_{|n|>m_f} |a_n(u^\rho_t)|^2 \to 0 \qquad \text{as } f \to \infty.$$

If the denominator in (68) is positive, this implies $\omega_f(u^\rho_t) \to 0$. If the denominator is zero, then $\omega_f(u^\rho_t) = 0$ for every f by convention. Hence $\omega_f(u^\rho_t) \to 0$ almost surely.

Since $0 \le \omega_f(\cdot) \le 1$, dominated convergence gives

$$\omega_{f,\rho}^{\text{pop}} = \mathbb{E}_{u_0,t}[\omega_f(u_t^\rho)] \to 0.$$

Because $w_1 > 0$, the quantity $w_1 \log \omega_{f,\rho}^{\text{pop}} + b_\omega$ tends to $-\infty$ whenever $\omega_{f,\rho}^{\text{pop}} > 0$ along the sequence, while the zero-tail convention covers the case $\omega_{f,\rho}^{\text{pop}} = 0$. Thus $\chi_{f,\rho} \to 0$, and $\lambda_{f,\rho} \to 0$.

Finally, $g_{t,\theta} = \chi_{f,\rho} g_{t,\theta}^{\text{cont}}$ and $0 \le g_{t,\theta}^{\text{cont}} \le 1$ entrywise. Hence

$$\frac{1}{SH}\|g_{t,\theta}\|_F^2 \le \chi_{f,\rho}^2 = \lambda_{f,\rho}.$$

Taking expectations proves the budget bound. □

G.6. Finite-sample Bounds

Following Assumption G.7, the independent sample unit is a trajectory. Let Z_i denote the i-th resolved trajectory, including histories and targets over the finite prediction horizon. For display, write a single-time loss; the finite-horizon average can be absorbed into the definition of ℓ_ψ.

For a predictor ψ, define

$$\ell_\psi(Z) := \left\| Y_t^{f,\rho} - \psi(\mathcal{U}_t^{f,\rho}) \right\|_{\mathcal{V}_f}^2. \tag{101}$$

For a predictor class Θ, define the induced loss class

$$\mathfrak{L}(\Theta) := \{Z \mapsto \ell_\psi(Z) : \psi \in \Theta\}. \tag{102}$$

We write

$$\mathfrak{L}_{f,\rho}^F := \mathfrak{L}(\mathcal{C}_{f,\rho}^F), \qquad \mathfrak{L}_{f,\rho}^{S4} := \mathfrak{L}(\mathcal{C}_{f,\rho}^{S4}), \tag{103}$$

$$\mathfrak{L}_{f,\rho}^{CG} := \mathfrak{L}(\mathcal{C}_{f,\rho}^{CG}), \qquad \mathfrak{L}_{f,\rho}^{AMG}(\lambda) := \mathfrak{L}(\mathcal{C}_{f,\rho}^{AMG}(\lambda)). \tag{104}$$

Let $R_m(\mathfrak{L})$ denote the population Rademacher complexity of a real-valued loss class $\mathfrak{L}$ with sample size m:

$$R_m(\mathfrak{L}) := \mathbb{E}_{Z_{1:m},\epsilon}\left[\sup_{\ell \in \mathfrak{L}} \frac{1}{m}\sum_{i=1}^m \epsilon_i \ell(Z_i)\right], \tag{105}$$

where ϵ_i are independent Rademacher random variables.

Under Assumption G.6, a standard Rademacher ERM bound implies that, with probability at least $1-\eta$, the empirical risk minimizer over any class Θ satisfies (Bach, 2024)

$$R_{f,\rho}(\widehat{\psi}^\Theta) \le \inf_{\psi \in \Theta} R_{f,\rho}(\psi) + 4R_m(\mathfrak{L}(\Theta)) + 2\ell_\infty\sqrt{\frac{\log(2/\eta)}{2m}}. \tag{106}$$

Applying this bound to the relevant classes and using Proposition G.1, define

$$U_{f,\rho}^F(\eta) := \varepsilon_{f,\rho}^2 + \mathcal{E}_{f,\rho}^F + 4R_m(\mathfrak{L}_{f,\rho}^F) + 2\ell_\infty\sqrt{\frac{\log(8/\eta)}{2m}}, \tag{107}$$

$$U_{f,\rho}^{S4}(\eta) := \varepsilon_{f,\rho}^2 + \mathcal{E}_{f,\rho}^{S4} + 4R_m(\mathfrak{L}_{f,\rho}^{S4}) + 2\ell_\infty\sqrt{\frac{\log(8/\eta)}{2m}}, \tag{108}$$

$$U_{f,\rho}^{CG}(\eta) := \varepsilon_{f,\rho}^2 + \mathcal{E}_{f,\rho}^{CG} + 4R_m(\mathfrak{L}_{f,\rho}^{CG}) + 2\ell_\infty\sqrt{\frac{\log(8/\eta)}{2m}}, \tag{109}$$

$$U_{f,\rho}^{AMG}(\lambda;\eta) := \varepsilon_{f,\rho}^2 + \mathcal{E}_{f,\rho}^{AMG}(\lambda) + 4R_m(\mathfrak{L}_{f,\rho}^{AMG}(\lambda)) + 2\ell_\infty\sqrt{\frac{\log(8/\eta)}{2m}}. \tag{110}$$

The common irreducible Bayes error and concentration terms cancel when subtracting the FFNO bound:

$$U^{S4}_{f,\rho}(\eta) - U^{F}_{f,\rho}(\eta) = -\Gamma^{S4}_{f,\rho} + 4\left(R_m(\mathfrak{L}^{S4}_{f,\rho}) - R_m(\mathfrak{L}^{F}_{f,\rho})\right), \tag{111}$$

$$U^{CG}_{f,\rho}(\eta) - U^{F}_{f,\rho}(\eta) = -\Gamma^{CG}_{f,\rho} + 4\left(R_m(\mathfrak{L}^{CG}_{f,\rho}) - R_m(\mathfrak{L}^{F}_{f,\rho})\right), \tag{112}$$

$$U^{AMG}_{f,\rho}(\lambda;\eta) - U^{F}_{f,\rho}(\eta) = -\Gamma^{AMG}_{f,\rho}(\lambda) + 4\left(R_m(\mathfrak{L}^{AMG}_{f,\rho}(\lambda)) - R_m(\mathfrak{L}^{F}_{f,\rho})\right). \tag{113}$$

These identities separate the approximation gain of memory from its effective finite-sample complexity cost.

Theorem G.11 (Low-resolution preference condition)**.** *At any fixed resolution f, a memory class has a smaller upper bound than FFNO whenever its approximation gain exceeds four times its Rademacher-complexity gap. In particular, S4FFNO satisfies $U^{S4}_{f,\rho}(\eta) < U^{F}_{f,\rho}(\eta)$ if*

$$\Gamma^{S4}_{f,\rho} > 4\left(R_m(\mathfrak{L}^{S4}_{f,\rho}) - R_m(\mathfrak{L}^{F}_{f,\rho})\right), \tag{114}$$

and AMGFNO satisfies $U^{AMG}_{f,\rho}(\lambda;\eta) < U^{F}_{f,\rho}(\eta)$ if

$$\Gamma^{AMG}_{f,\rho}(\lambda) > 4\left(R_m(\mathfrak{L}^{AMG}_{f,\rho}(\lambda)) - R_m(\mathfrak{L}^{F}_{f,\rho})\right). \tag{115}$$

Proof. Both statements follow directly from (111) and (113). If the gain is larger than four times the complexity gap, the corresponding upper-bound difference is negative. □

G.7. A Sufficient Condition for AMGFNO Complexity Convergence

This subsection gives a sufficient condition under which the AMGFNO loss-class complexity returns to the FFNO loss-class complexity as the spectral gate closes. The condition is not necessary; it makes the conditional finite-sample comparison explicit.

Use the normalized feature norm

$$\|A\|^2_{S,H} := \frac{1}{SH}\|A\|^2_F. \tag{116}$$

Proposition G.12 (AMGFNO complexity convergence under a closing gate)**.** *Suppose there exist constants $L_{\mathrm{Dec}}, B_h, B_z, B_o < \infty$, independent of f, such that for all considered parameters and samples,*

$$\|\mathrm{Dec}_\theta(a) - \mathrm{Dec}_\theta(b)\|_{\mathcal{V}_f} \le L_{\mathrm{Dec}}\|a-b\|_{S,H}, \tag{117}$$

$$\|h_{t,\theta}\|_{S,H} \le B_h, \qquad \|z_{t,\theta}\|_{S,H} \le B_z, \tag{118}$$

$$\|Y^{f,\rho}_t\|_{\mathcal{V}_f} \le B_o, \qquad \|\psi(\mathcal{U}^{f,\rho}_t)\|_{\mathcal{V}_f} \le B_o \quad \textit{for all predictors considered.} \tag{119}$$

Then, for $\lambda_{f,\rho} = \chi^2_{f,\rho}$,

$$0 \le R_m(\mathfrak{L}^{AMG}_{f,\rho}(\lambda_{f,\rho})) - R_m(\mathfrak{L}^{F}_{f,\rho}) \le C_{\mathrm{comp}}\sqrt{\lambda_{f,\rho}}, \tag{120}$$

where C_{comp} is independent of f. Consequently, this complexity gap converges to zero as $f \to \infty$.

Proof. The lower bound follows from monotonicity of Rademacher complexity because Proposition G.2 gives $\mathcal{C}^{F}_{f,\rho} \subset \mathcal{C}^{AMG}_{f,\rho}(\lambda_{f,\rho})$ in closure, and therefore $\mathfrak{L}^{F}_{f,\rho} \subset \mathfrak{L}^{AMG}_{f,\rho}(\lambda_{f,\rho})$ in closure.

For the upper bound, take any $\psi^{AMG}_\theta \in \mathcal{C}^{AMG}_{f,\rho}(\lambda_{f,\rho})$ and define its FFNO fallback with the same encoder, pre-AMG layers, and decoder:

$$\psi^{F}_\theta(\mathcal{U}^{f,\rho}_t) := \mathrm{Dec}_\theta(h_{t,\theta}). \tag{121}$$

The AMGFNO fused feature differs from this fallback feature by

$$\begin{aligned}(1-g_{t,\theta}) \odot h_{t,\theta} + g_{t,\theta} \odot z_{t,\theta} - h_{t,\theta} \\ = g_{t,\theta} \odot (z_{t,\theta} - h_{t,\theta}).\end{aligned} \tag{122}$$

Since $g_{t,\theta} = \chi_{f,\rho} g^{\mathrm{cont}}_{t,\theta}$ and $0 \le g^{\mathrm{cont}}_{t,\theta} \le 1$ entrywise,

$$\|g_{t,\theta}\|_\infty \le \chi_{f,\rho} = \sqrt{\lambda_{f,\rho}}.$$

Therefore, using (118),

$$\begin{aligned}\|g_{t,\theta} \odot (z_{t,\theta} - h_{t,\theta})\|_{S,H} &\le \sqrt{\lambda_{f,\rho}}\|z_{t,\theta} - h_{t,\theta}\|_{S,H} \\ &\le \sqrt{\lambda_{f,\rho}}(B_h + B_z).\end{aligned} \tag{123}$$

By the decoder Lipschitz condition (117),

$$\left\|\psi_\theta^{AMG}(\mathcal{U}_t^{f,\rho}) - \psi_\theta^{F}(\mathcal{U}_t^{f,\rho})\right\|_{\mathcal{V}_f} \le L_{\mathrm{Dec}}(B_h + B_z)\sqrt{\lambda_{f,\rho}}. \tag{124}$$

For any target y and predictions p, q with norms bounded by B_o,

$$\begin{aligned}\left|\|y-p\|^2 - \|y-q\|^2\right| &= |\langle q-p, 2y-p-q\rangle| \\ &\le \|p-q\|\,(2\|y\| + \|p\| + \|q\|) \\ &\le 4B_o\|p-q\|,\end{aligned} \tag{125}$$

where the last step uses (119). Combining this with (124), there exists

$$C_{\mathrm{loss}} := 4B_o L_{\mathrm{Dec}}(B_h + B_z)$$

such that

$$\left|\ell_{\psi_\theta^{AMG}}(Z) - \ell_{\psi_\theta^{F}}(Z)\right| \le C_{\mathrm{loss}}\sqrt{\lambda_{f,\rho}} \tag{126}$$

for every sample Z.

Define the difference class

$$\mathfrak{D}_{f,\rho} := \left\{Z \mapsto \ell_{\psi_\theta^{AMG}}(Z) - \ell_{\psi_\theta^{F}}(Z) : \psi_\theta^{AMG} \in \mathcal{C}_{f,\rho}^{AMG}(\lambda_{f,\rho})\right\}. \tag{127}$$

By (126), every function in $\mathfrak{D}_{f,\rho}$ has supremum norm at most $C_{\mathrm{loss}}\sqrt{\lambda_{f,\rho}}$, so

$$R_m(\mathfrak{D}_{f,\rho}) \le C_{\mathrm{loss}}\sqrt{\lambda_{f,\rho}}.$$

Each AMGFNO loss can be written as an FFNO fallback loss plus a function in $\mathfrak{D}_{f,\rho}$. By subadditivity of Rademacher complexity,

$$\begin{aligned}R_m(\mathfrak{L}_{f,\rho}^{AMG}(\lambda_{f,\rho})) &\le R_m(\mathfrak{L}_{f,\rho}^{F}) + R_m(\mathfrak{D}_{f,\rho}) \\ &\le R_m(\mathfrak{L}_{f,\rho}^{F}) + C_{\mathrm{loss}}\sqrt{\lambda_{f,\rho}}.\end{aligned} \tag{128}$$

Taking $C_{\mathrm{comp}} = C_{\mathrm{loss}}$ proves the upper bound. Since $\lambda_{f,\rho} \to 0$ by Lemma G.10, the gap converges to zero. The subadditivity and bounded-difference steps are standard Rademacher-complexity arguments (Bach, 2024). □

G.8. Proof of Theorem 2.2

Proof of Theorem 2.2. By Lemma G.10,

$$\lambda_{f,\rho} = \chi_{f,\rho}^2 \to 0.$$

By Proposition G.12,

$$0 \le R_m(\mathfrak{L}_{f,\rho}^{AMG}(\lambda_{f,\rho})) - R_m(\mathfrak{L}_{f,\rho}^{F}) \le C_{\mathrm{comp}}\sqrt{\lambda_{f,\rho}} \to 0. \tag{129}$$

From the finite-sample decomposition in (113),

$$\begin{aligned}&U_{f,\rho}^{AMG}(\lambda_{f,\rho};\eta) - U_{f,\rho}^{F}(\eta) \\ &\quad = -\Gamma_{f,\rho}^{AMG}(\lambda_{f,\rho}) + 4\left(R_m(\mathfrak{L}_{f,\rho}^{AMG}(\lambda_{f,\rho})) - R_m(\mathfrak{L}_{f,\rho}^{F})\right).\end{aligned} \tag{130}$$

Theorem 2.1 gives

$$0 \le \Gamma_{f,\rho}^{AMG}(\lambda_{f,\rho}) \le \mathcal{E}_{f,\rho}^{F} \to 0.$$

Together with (129), this proves

$$U^{AMG}_{f,\rho}(\lambda_{f,\rho};\eta) - U^{F}_{f,\rho}(\eta) \to 0.$$

For fixed-fusion S4FFNO, (111) gives

$$U^{S4}_{f,\rho}(\eta) - U^{F}_{f,\rho}(\eta) = -\Gamma^{S4}_{f,\rho} + 4\left(R_m(\mathfrak{L}^{S4}_{f,\rho}) - R_m(\mathfrak{L}^{F}_{f,\rho})\right). \tag{131}$$

Again, Theorem 2.1 gives

$$0 \le \Gamma^{S4}_{f,\rho} \le \mathcal{E}^{F}_{f,\rho} \to 0.$$

If the fixed-fusion S4FFNO complexity gap satisfies

$$R_m(\mathfrak{L}^{S4}_{f,\rho}) - R_m(\mathfrak{L}^{F}_{f,\rho}) \to \Delta^{S4}_{m,\rho} > 0,$$

then

$$U^{S4}_{f,\rho}(\eta) - U^{F}_{f,\rho}(\eta) \to 4\Delta^{S4}_{m,\rho} > 0.$$

At the same time,

$$U^{AMG}_{f,\rho}(\lambda_{f,\rho};\eta) - U^{F}_{f,\rho}(\eta) \to 0.$$

Therefore, for all sufficiently large f,

$$U^{AMG}_{f,\rho}(\lambda_{f,\rho};\eta) < U^{S4}_{f,\rho}(\eta).$$

□

G.9. Interpretation and Scope

The theory compares FFNO, S4FFNO, and AMGFNO as predictors defined on the same resolved history $\mathcal{U}^{f,\rho}_t$. FFNO is the restricted subclass that ignores history, S4FFNO uses fixed residual memory fusion, and AMGFNO uses the same S4 memory branch with the adaptive memory gate.

Theorem 2.1 isolates the memory benefit induced by projection-based loss of Fourier modes. At low resolution, unresolved spatial modes can still influence the next resolved state, so the history-conditioned Bayes target may depend on temporal history. This is the condition in which the low-resolution preference condition in Theorem G.11 can favor memory models. At high resolution, under the stated projection-induced partial observation assumptions, the unresolved tail $Q_f u^{\rho}_t$ vanishes and the projected Markovian target becomes sufficient. The additional approximation gain attributable to this spectral-projection mechanism therefore converges to zero.

Theorem 2.2 gives a conditional finite-sample comparison. Fixed-fusion S4FFNO may retain a nonvanishing effective complexity cost, while AMGFNO drives its gate budget $\lambda_{f,\rho}$ to zero through the frequency-aware gate. Under the sufficient conditions of Proposition G.12, the AMGFNO Rademacher-complexity gap is $O(\sqrt{\lambda_{f,\rho}})$ and therefore vanishes in the high-resolution limit.

The analysis controls one-step resolved prediction risk. It does not directly bound accumulated autoregressive rollout error, which is evaluated empirically in Section 4. The analysis also isolates only the spectral projection mechanism. It does not rule out persistent memory benefits from other sources of partial observability, such as latent parameters, stochastic forcing, measurement noise, temporal subsampling, or unresolved physical variables. This is consistent with the Mori–Zwanzig viewpoint that unresolved variables can induce non-Markovian closure terms (Parish & Duraisamy, 2017).