MDPI

Article

Filter Before Mixing: Per-Modality Denoising for Multimodal RL with Application to Health Management

Tsuyoshi Okita[1]

[1] Kyushu Institute of Technology; tsuyoshi@ai.kyutech.ac.jp
* Correspondence: tsuyoshi@ai.kyutech.ac.jp

Abstract

Multimodal reinforcement learning agents must fuse signals with vastly different noise profiles—yet existing architectures, whether monolithic (π0, DreamerV3) or modular (MSDP, VTDexManip), allow noise from unreliable modalities to contaminate reliable ones at the point of fusion. We propose *filter-before-mixing*: each modality's representation is independently refined by a per-modality Flow Matching module before spectral-domain fusion via a Fourier Neural Operator (FNO), with a residual gate ensuring that refinement is never harmful. The resulting architecture, **FreamerV1** (Filter-before-mixing dreamer), has 93M parameters (0.4M trainable). On MiniGrid, FreamerV1 reaches 87.7% $\pm$ 8.2% (3 seeds) at 5000 episodes, while the encoder-only baseline degrades to 78% due to catastrophic forgetting. With OGM-GE (On-the-fly Gradient Modulation) for adaptive per-modality gate control, FreamerV1 achieves an 8.0% relative improvement in success rate over manual tuning with halved seed-to-seed variance (3 seeds). On Crafter (no language modality), it achieves an 11.7% relative improvement over DreamerV3 in the official Crafter score (geometric mean of 22 achievement success rates; 10 seeds). On PAMAP2 wearable sensors—where no pre-trained encoder exists—the foundation encoder achieves 2.4$\times$ higher reward and 16$\times$ lower variance than a vanilla MLP, confirming that the filter-before-mixing advantage grows with encoder noise.

Keywords: multimodal reinforcement learning, per-modality denoising, flow matching, Fourier neural operator, spectral fusion, slot attention, catastrophic forgetting, wearable health management, world model, episodic memory

1. Introduction

Reinforcement learning agents operating in the physical world receive information through multiple sensors: cameras capture spatial structure, proprioceptive sensors report joint angles, language instructions convey goals, and reward signals provide sparse feedback. These modalities differ vastly in noise level, sampling rate, and information density. A robust multimodal RL system must combine them without allowing noise from one channel to degrade the others.

Recent RL architectures address multimodal fusion in two broad ways. *Monolithic* approaches process all modalities through a single shared backbone. DreamerV3 [1] feeds observations through a shared encoder–RSSM pipeline with fixed hyperparameters, mastering over 150 tasks spanning Atari to robotic control. π0 [2] tokenizes vision, language, state, and action noise into a 3B-parameter PaliGemma transformer with flow matching for continuous action generation. Genie 2 [3] and DIAMOND [4] apply diffusion to world

Received:
Revised:
Accepted:
Published:

Copyright: © 2026 by the authors. Submitted to *Journal Not Specified* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license.

model prediction and video generation respectively. These systems rely on model scale to implicitly absorb distributional differences across modalities.

Modular approaches assign a dedicated encoder to each modality, then fuse the outputs. MSDP [5] pretrains a multisensory encoder via masked autoencoding across vision, force, and proprioception, then fuses embeddings through cross-attention. VTDexManip [6] concatenates CLIP visual features with tactile MLP features for dexterous manipulation, finding that adding sparse touch signals improves success rates by 20%. MIB [7] compresses a joint representation via the information bottleneck principle, filtering task-irrelevant information after fusion. These methods use separate encoders but do not equalize signal quality before fusion: the noisy output of a sparse reward encoder enters the same fusion layer as a well-encoded CLIP embedding.

Both paradigms thus share a common vulnerability: *cross-modal noise contamination* at the point of fusion. When a high-variance modality (e.g., sparse reward, raw IMU acceleration) is fused with a low-variance one (e.g., CLIP vision embedding), noise from the former can corrupt the latter. This problem is especially acute in sensor-driven domains such as wearable health management, where modalities have radically different noise profiles and temporal scales—physical movement precedes heart-rate elevation by several seconds, and language instructions precede visual confirmation by many steps.

A second gap is the *absence of principled spectral-domain fusion*. Standard attention computes instantaneous pairwise similarities and cannot naturally represent temporally delayed cross-modal correlations. A third gap is the *limited application to healthcare*. MedDreamer [8] applied an RSSM to electronic health records, and KANDI [9] used diffusion policies for elderly activity promotion, but no prior work has combined per-modality generative refinement with spectral fusion for wearable-sensor health intervention.

This paper addresses these gaps through a design principle we call *filter-before-mixing*: each modality's representation is independently refined by a dedicated Flow Matching module before cross-modal fusion, analogous to the signal processing practice of filtering each channel before mixing into a master bus. The refinement intensity adapts to each modality's signal quality: weak for already well-encoded signals (e.g., frozen CLIP embeddings) and strong for noisy or sparse signals (e.g., raw reward, IMU data). The refined representations are then fused in the spectral domain via a Fourier Neural Operator (FNO), whose complex-valued weights naturally encode phase-shifted cross-modal correlations. An information-theoretic analysis (Section 3.3) shows that pre-fusion denoising preserves more policy-relevant mutual information than post-fusion alternatives, with a residual gate mechanism ensuring that the refinement is never harmful.

The resulting four-layer modular architecture integrates modality-specific encoding (frozen CLIP with Slot Attention, or IMU foundation encoders for health applications), per-modality Flow Matching, FNO spectral fusion, and DNC episodic memory. The modular design enables domain transfer: the same downstream pipeline (Flow Matching $\rightarrow$ FNO $\rightarrow$ DNC $\rightarrow$ PPO) is shared between MiniGrid navigation and PAMAP2 health management, with only the modality-specific encoders replaced.

We validate the framework in three domains. In *MiniGrid navigation*, the system achieves 94.0% success on MultiRoom-N2-S4 (+51 pp over IMPALA under identical conditions) and 93.0% on N4-S4, while IMPALA fails entirely on harder configurations (0% on N4-S5 and N6). In *Crafter*, a procedurally generated open-world environment with no language instructions, the agent achieves an official score of $16.2\% \pm 0.8\%$ (10 seeds), exceeding DreamerV3 (14.5%) despite operating without the language modality. In *wearable-sensor health management* on the PAMAP2 dataset—where no pre-trained encoder exists and per-modality denoising is critical—the foundation encoder achieves $2.4\times$ higher cumulative reward (268.3 ± 12.0) and $16\times$ lower cross-seed variance than a vanilla MLP

Table 1. Design Differentiation from $\pi 0$. The two systems target fundamentally different regimes: $\pi 0$ addresses continuous-control robotic manipulation with large-scale demonstration data, while our framework targets discrete-action sensor-driven domains with limited data.

Design Aspect	$\pi 0$	FreamerV1 (ours)
Target domain	Robotic manipulation	Sensor-driven decision
Architecture	Monolithic VLM	Modular per-modality
FM target	Actions (output)	Representations (internal)
Cross-modal fusion	Self-attention	FNO (spectral domain)
Action space	Continuous	Discrete
External memory	None	DNC
Parameters	3B	93M (0.4M trainable)

baseline. This progression—MiniGrid (4 modalities, CLIP available), Crafter (3 modalities, no language), PAMAP2 (3 modalities, no pre-trained encoder)—illustrates a key insight: the advantage of filter-before-mixing grows with encoder noise and is robust to missing modalities.

The contributions of this paper are as follows:

1. We propose the *filter-before-mixing* design principle for multimodal RL, in which per-modality Flow Matching denoises each representation before spectral-domain fusion, and provide an information-theoretic justification with an explicit approximation bound (Eq. (A18)).
2. We instantiate this principle in FreamerV1 (Filter-before-mixing dreamer), a modular four-layer architecture (93M parameters, 0.4M trainable), and validate it on MiniGrid navigation with controlled baselines (IMPALA under identical conditions, PPO).
3. We validate across three domains—MiniGrid (4 modalities), Crafter (3 modalities, no language), and PAMAP2 (3 modalities, no pre-trained encoder)—showing that the architecture degrades gracefully with missing modalities and that the filter-before-mixing advantage grows with encoder noise.

Table 1 summarizes how these design choices differentiate FreamerV1 from $\pi 0$. These are not claims of superiority; they reflect optimization for a different regime—sensor-driven decision-making with discrete actions, limited data, and the need for interpretable, modular policies.

2. Related Work

We organize related work along the three problems identified in the Introduction: (1) how existing architectures fuse multimodal inputs and where cross-modal noise contamination arises, (2) how generative models have been applied in RL and where our per-modality approach diverges, and (3) how world models have been used in healthcare.

2.1. Multimodal Fusion in RL Foundation Models

The dominant approach to multimodal fusion in recent RL foundation models is *monolithic*: all modalities are tokenized and processed by a single large transformer. $\pi 0$ [2] feeds image tokens (from PaliGemma), language tokens, robot-state tokens, and noisy action tokens into a shared self-attention backbone, relying on the model's 3B-parameter capacity to implicitly disentangle heterogeneous inputs. Gato [10] similarly serializes text, images, and actions into a single token sequence for a 1.2B-parameter transformer. DreamerV3 [1] fuses observations through a shared encoder–RSSM pipeline with fixed hyperparameters across both discrete and continuous action domains, achieving human-level Atari performance and competitive robotic control.

These architectures have proven effective when the input modalities are relatively homogeneous (e.g., images and proprioception in robotic manipulation) or when the model is large enough to absorb distributional differences. However, they do not explicitly prevent noise from one modality from propagating to another during fusion—the problem we term *cross-modal noise contamination*. We address this by denoising each modality independently via per-modality Flow Matching before fusion.

A related line of work explores modality-specific processing. CLIP [11] demonstrated the power of separate vision and language encoders aligned through contrastive learning. Slot Attention [12] decomposes visual inputs into object-centric slots, enabling structured representation learning. SAVi [13] extended this to video with temporally consistent slot tracking. We adopt both CLIP and Slot Attention as modality-specific encoders in Layer 1 of our architecture, using them as established building blocks rather than claiming novelty for their individual designs.

2.1.1. Multimodal Sensor Fusion in RL

Recent work has increasingly recognized that naive fusion of heterogeneous sensor modalities can degrade RL performance. Meng et al. [7] demonstrated that concatenating egocentric images and proprioception can fail to match single-modality performance, and proposed a Multimodal Information Bottleneck (MIB) that compresses joint representations while retaining task-relevant information. Their approach filters out task-irrelevant information after fusion, in contrast to our per-modality pre-fusion denoising.

In contact-rich robotic manipulation, MSDP [5] proposed self-supervised multisensory pretraining using masked autoencoding across vision, force, and proprioception. Their key insight is an asymmetric actor-critic architecture: the critic uses cross-attention over frozen sensor embeddings for dynamic feature extraction, while the actor receives a stable pooled representation. This achieves robustness to sensor noise with as few as 6,000 online interactions on real hardware. However, MSDP does not apply modality-specific noise reduction before fusion—all sensor embeddings enter the same transformer encoder and are masked uniformly.

VTDexManip [6] presented a benchmark for visual-tactile dexterous manipulation, comparing 17 pretrained and non-pretrained methods. Their results showed that adding sparse binary tactile signals to vision improves success rates by approximately 20%, and joint visual-tactile pretraining gains a further 20%. Fusion is achieved by concatenating visual features (from CLIP, R3M, or ResNet) with tactile MLP features—a simple strategy that does not account for the very different noise characteristics of high-resolution images versus sparse binary contact signals.

These approaches share a common pattern: modality-specific encoders followed by concatenation, cross-attention, or information-theoretic fusion. None applies generative refinement (e.g., Flow Matching) to individual modality representations before fusion. Our per-modality FM fills this gap by allowing the refinement intensity to be tuned per modality—weak for already well-encoded signals (e.g., CLIP vision) and strong for noisy or sparse signals (e.g., reward, raw IMU)—before spectral fusion combines the cleaned representations. In the supervised learning literature, noise-robust training has been addressed through sample selection (e.g., PSSCL [14], which uses contrastive loss to identify clean samples under label noise). Our approach differs fundamentally: rather than selecting clean samples, we denoise the *representations themselves* at the modality level before fusion, which is applicable even when no clean/noisy partition of the data exists.

2.2. Generative Models and Memory in RL

Generative models in RL have been applied almost exclusively at the *output* stage: Decision Diffuser [15] generates trajectories, Diffusion Policy [16] produces action chunks, $\pi 0$ [2] applies flow matching to 50-step action generation, and DIAMOND [4] learns world models via diffusion over observations. All operate *after* multimodal fusion. Our approach inverts this: Flow Matching operates on internal representations *before* fusion, refining each modality's encoding via learned optimal transport. For fusion itself, we employ a Fourier Neural Operator (FNO) [17], whose complex-valued spectral weights naturally encode phase-shifted cross-modal correlations—a property absent from attention-based fusion.

For episodic memory, the Differentiable Neural Computer (DNC) [18] provides content-based external memory, previously applied to RL navigation in MERLIN [19]. In our framework, the DNC restores episodic context that is lost when the FNO produces a single-timestep fused representation.

2.3. World Models: From Games to Healthcare

World models—learned environment simulators that predict future states from actions—have become a dominant paradigm for sample-efficient RL. DreamerV2 [20] introduced categorical latent representations for discrete-action Atari games. DreamerV3 [1] generalized this across 150+ tasks with fixed hyperparameters, spanning Atari, DMControl, Minecraft, and Crafter, handling both discrete and continuous actions. MuZero [21] combined learned models with Monte Carlo tree search for board games and Atari. More recently, DIAMOND [4] and Genie 2 [3] have explored diffusion-based world models at foundation scale.

Despite this success, world model-based RL remains underexplored in healthcare. MedDreamer [8] is the closest precedent: it adapts the DreamerV3 RSSM to electronic health records (EHRs) for sepsis treatment and mechanical ventilation, introducing an Adaptive Feature Integration module to handle irregular clinical time series. KANDI [9] addresses physical activity promotion for fall-risk elderly using wearable accelerometers, combining Diffusion Policies with offline inverse RL. However, neither MedDreamer nor KANDI employs structured multimodal encoding (Slot Attention, per-modality Flow Matching) or spectral-domain fusion, and neither targets continuous wearable-sensor health management with online imagination-based policy optimization.

2.4. Positioning of This Work

Table 2 positions our framework against existing methods along five capability axes that correspond to the design choices motivated in the Introduction. No prior method combines per-modality generative representation refinement, spectral-domain cross-modal fusion, language-grounded reward shaping, external episodic memory, and world model-based health intervention.

3. Proposed Method

This section describes the architecture that instantiates the filter-before-mixing principle introduced in Section I.

3.1. Architecture Overview

The architecture consists of four layers (Fig. 1):

1. **Layer 1 — Modality-Specific Encoding.** Each of four input modalities (state, vision, language, reward) is encoded by a dedicated module. Vision uses a frozen CLIP ViT-B/32 [11] with Slot Attention [12]; language uses a Transformer encoder [22] with

Table 2. Capability Comparison with Related Methods

Method	Per-modal Gen. Enc.	Spectral Fusion	Language Reward	External Memory	Health WM	Scale >1B	Cont. Ctrl
DreamerV3	–	–	–	–	–	–	◯
$\pi 0$	–	–	–	–	–	◯	◯
DIAMOND	–	–	–	–	–	–	–
Decision Diff.	–	–	–	–	–	–	◯
Diffusion Policy	–	–	–	–	–	–	◯
MERLIN	–	–	–	◯	–	–	◯
MedDreamer	–	–	–	–	◯	–	–
KANDI	–	–	–	–	◯	–	–
FreamerV1 (ours)	◯	◯	◯	◯	◯	–	–

Per-modal Gen. Enc.: per-modality generative encoding (FM on representations). Spectral Fusion: FNO-based cross-modal fusion. Language Reward: LLM-based dense reward shaping. External Memory: DNC or equivalent. Health WM: world model applied to health intervention. Scale >1B: model with >1B parameters. Cont. Ctrl: validated on continuous-control tasks.

Slot Attention; state uses an FNO-based encoder [17]; reward uses a temporal Conv1D encoder. All encoders produce d-dimensional embeddings.

2. **Layer 2 — Per-Modality Flow Matching.** Each modality embedding is refined by an independent Flow Matching module [23] that learns an optimal transport map from a Gaussian prior to the modality's data manifold, denoising the representation before cross-modal fusion.
3. **Layer 3 — FNO Spectral Fusion.** The refined modality embeddings are stacked as a $M \times d$ matrix and fused via spectral-domain convolution [17] (SpectralConv1d), producing a unified representation.
4. **Layer 4 — DNC Memory + Policy.** The fused representation is augmented with episodic context from a DNC memory module [18], and fed to a PPO [24] policy head that produces a discrete action.

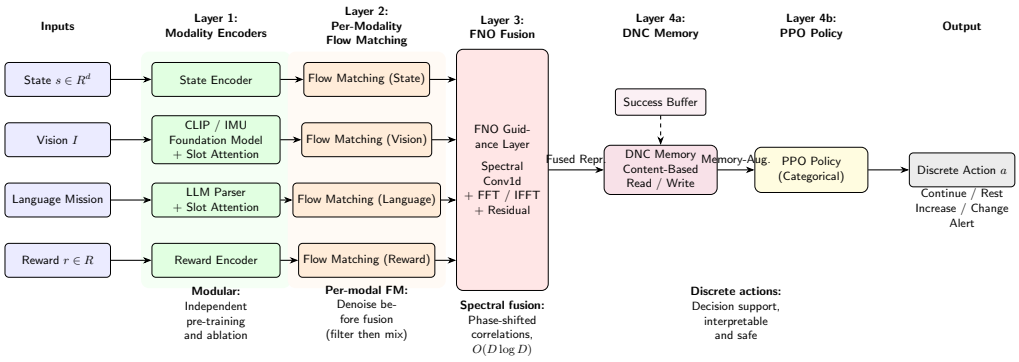

Figure 1. Overall Architecture of the Proposed Method

The total parameter count is 93.05M, of which 92.68M (99.6%) are the frozen CLIP encoder. The trainable parameters amount to 0.37M, enabling efficient learning on small datasets.

3.2. Layer 1: Modality-Specific Encoding

Each modality is encoded by a dedicated module producing a d-dimensional embedding (details in Appendix E).

For vision, a frozen CLIP ViT-B/32 [11] extracts 49 patch features (7×7 grid, 768-dim), projected to $\mathbb{R}^d$. Slot Attention [12] decomposes the features into K object-centric slots via

iterative competitive assignment. Four numerical stabilization techniques reduce the NaN occurrence rate from 23.5% to 0% (Table 5).

For language, the mission text is processed by a Transformer encoder and decomposed into K_{lang} slots via Slot Attention. An LLM (Qwen2.5 [25]) parses the mission into structured components for reward shaping (Appendix A).

For state, the MiniGrid observation ($7 \times 7 \times 3$) is flattened and encoded by a 1D FNO block [17].

For reward, the past H-step reward history is encoded by a temporal Conv1D.

Bidirectional multi-head cross-modal attention between visual and language slots enables component-level correspondence (e.g., "green goal" slot $\leftrightarrow$ green object slot). The resulting slots are mean-pooled to produce modality embeddings $e_m \in \mathbb{R}^d$.

3.3. Layer 2: Per-Modality Flow Matching

For each modality $m \in \{\text{state, vision, language, reward}\}$, we apply an independent Flow Matching module ϕ^m to refine the encoded representation e_m before cross-modal fusion. This implements the "filter-before-mixing" principle (Section III-A-2).

Each module learns a velocity field v_θ^m via Conditional Flow Matching (CFM) [23] and refines e_m by Euler integration from $t = 0$ to $t = 1$ over N steps (see Appendix E for the full formulation):

$$x_0^m = e_m \tag{1}$$

$$x_{i+1}^m = x_i^m + \frac{1}{N} v_\theta^m(x_i^m, i/N), \quad i = 0, \ldots, N-1 \tag{2}$$

A learnable residual gate ensures training stability:

$$\hat{e}_m = e_m + \sigma(\alpha^m) \cdot (x_N^m - e_m) \tag{3}$$

where α^m is initialized to -3 so that $\sigma(\alpha^m) \approx 0.047$ at the start of training, ensuring that the FM module acts as a near-identity function until its velocity field is sufficiently trained. The refinement intensity can be set independently per modality: weak for already well-encoded signals (e.g., frozen CLIP: $\alpha = -5$, $\sigma \approx 0.007$) and strong for noisy signals (e.g., sparse reward: $\alpha = -1$, $\sigma \approx 0.27$). Rather than tuning α^m manually, we adopt OGM-GE gradient modulation [26] to adaptively scale the α^m gradients (with $10\times$ learning rate amplification) based on each modality's FM loss convergence rate, improving mean performance by +7 pp and halving seed-to-seed variance (Section 4.3.5).

We provide an information-theoretic argument for why applying Flow Matching to each modality independently *before* fusion is preferable to applying it after fusion (see Appendix L for the full derivation).

Let $\{X^m\}_{m=1}^M$ denote the encoded representations of M modalities, each corrupted by modality-specific noise: $\tilde{X}^m = X^m + \epsilon^m$, where $\epsilon^m \sim \mathcal{N}(0, \sigma_m^2 I)$ and the noise variances σ_m^2 differ across modalities.

In *post-fusion denoising* (as in $\pi 0$), the noisy representations are first fused as $Z = f(\tilde{X}^1, \ldots, \tilde{X}^M)$ and then denoised. By the data processing inequality [27], information lost during noisy fusion cannot be recovered. Cross-modal noise propagates through shared projection weights.

In *pre-fusion denoising* (our approach), a modality-specific denoiser g^m is applied to obtain $\hat{X}^m \approx X^m$, then the cleaned representations are fused. If each FM achieves near-optimal denoising:

$$I\big((X^1, \ldots, X^M);\, f(\hat{X}^1, \ldots, \hat{X}^M)\big) \geq I\big((X^1, \ldots, X^M);\, f(\tilde{X}^1, \ldots, \tilde{X}^M)\big) \tag{4}$$

In practice, the FM modules are imperfect denoisers with error $\delta^m = \hat{X}^m - \mathbb{E}[X^m|\tilde{X}^m]$. Pre-fusion denoising remains beneficial whenever $\|\delta^m\|^2 < \sigma_m^2$—a condition much weaker than optimal denoising (see Appendix L for the full derivation). The residual gate further tightens this bound: since $\sigma(\alpha^m) \to 0$ when the velocity field is untrained, pre-fusion denoising is *never worse* than no denoising.

The above argument relies on two key assumptions: (i) modality-specific noise is approximately Gaussian, and (ii) FM achieves denoising error $\|\delta^m\|^2 < \sigma_m^2$. We assess these for each domain: In **MiniGrid**, the frozen CLIP encoder produces near-deterministic embeddings for any given frame ($\sigma^2_{\text{vision}} \approx 0$), while the sparse reward signal has high variance ($\sigma^2_{\text{reward}} \gg 0$) due to rare success events. The Gaussian assumption holds approximately because the noise arises from the stochastic policy's action sampling rather than sensor noise. In **Crafter**, vision noise increases due to procedural generation, and the absence of language removes one modality entirely, but the remaining three satisfy the same conditions. In **PAMAP2**, raw IMU and heart-rate signals have well-characterized sensor noise profiles that are approximately Gaussian, and no pre-trained encoder is available to reduce σ_m^2—precisely the regime where condition (ii) is most easily satisfied and the benefit largest. The noise injection experiment (Table 13) empirically validates that FM preserves performance under σ up to 3.0, confirming that condition (ii) holds in practice.

3.4. Layer 3: FNO Spectral Fusion

The four refined modality embeddings $\hat{e}_m \in \mathbb{R}^d$ are stacked into a matrix $E \in \mathbb{R}^{M\times d}$ and fused via a Fourier Neural Operator (FNO) [17]. The FNO applies spectral convolution: FFT along the embedding dimension, multiplication by learnable complex-valued weights $R_k \in \mathbb{C}^{M\times M}$ for the first K Fourier modes, and inverse FFT, with a pointwise residual path (see Appendix E for the full formulation). The fused representation is obtained by mean-pooling over the modality axis.

The key advantage of spectral-domain fusion over attention-based fusion is the ability to represent *phase-shifted cross-modal correlations*. When two modalities have a temporal delay τ in their correlation (e.g., IMU activity precedes heart-rate elevation), this appears as a frequency-dependent phase shift $e^{-j\omega\tau}$ in the Fourier domain. The FNO's complex-valued weights can directly encode such shifts through $\arg(R_k)$, whereas real-valued attention computes instantaneous inner products and cannot represent phase delays without auxiliary mechanisms.

It is important to distinguish the role of the FNO here from the FNO-based state encoder in Layer 1. The Layer 1 FNO encodes a single modality (the grid observation) into a d-dimensional embedding—a standard spectral encoding operation. The Layer 3 FNO, by contrast, operates *after* per-modality Flow Matching and serves a fundamentally different purpose: it acts as a PDE-guided fusion operator over the denoised representations. When different modalities undergo different FM refinement trajectories (because their velocity fields and gate values differ), the resulting representations lie on different regions of the learned manifold. The Layer 3 FNO resolves these heterogeneous trajectories into a single coherent representation by leveraging the spectral structure of the cross-modal correlations—analogous to how an FNO solves a PDE by operating in the frequency domain over spatially heterogeneous initial conditions. This PDE-guidance role becomes increasingly important as the diversity of FM trajectories grows, which occurs when modalities have substantially different noise levels (as in PAMAP2, where raw IMU and heart-rate signals follow very different refinement paths) or when the environment requires the agent to integrate temporally delayed cross-modal signals.

3.5. Layer 4: DNC Memory and PPO Policy

The fused representation e_{fused} is augmented with episodic context via a Differentiable Neural Computer (DNC) [18] with content-based addressing, producing a memory-augmented representation e_{aug}. A categorical PPO [24] policy head then produces discrete actions:

$$\pi_\theta(a \mid s) = \text{softmax}(W_\pi\, e_{\text{aug}} + b_\pi) \tag{5}$$

The DNC architecture details (content-based addressing, write/read operations) and PPO objective formulation are provided in Appendix F.

3.6. Auxiliary Components

The framework includes several auxiliary mechanisms whose details are in Appendix G: *Language-grounded reward shaping* provides dense supervision in sparse-reward environments by matching LLM-parsed mission structure against observations [25]. *Adaptive reward shaping* [28] decays bonus rewards as the success rate increases. *Success Buffer* [29] stores and replays successful episodes to prevent catastrophic forgetting. *Score-based adaptive complexity estimation* [30] dynamically adjusts the number of FM inference steps per modality.

3.7. Training Objective

The overall loss function combines four terms:

$$\mathcal{L} = \mathcal{L}_{\text{PPO}} + \lambda_{\text{FM}}\mathcal{L}_{\text{FM}} + \lambda_{\text{CL}}\mathcal{L}_{\text{CL}} + \lambda_s\mathcal{L}_{\text{smooth}} \tag{6}$$

where $\mathcal{L}_{\text{FM}}$ is the per-modality Flow Matching loss (Eq. (A13)), $\mathcal{L}_{\text{CL}}$ is a vision–language contrastive alignment loss [11], and $\mathcal{L}_{\text{smooth}}$ is the FNO smoothness regularization (Eq. (A16)).

The total loss combines four objectives operating at different levels of the architecture: $\mathcal{L}_{\text{FM}}$ trains the per-modality Flow Matching velocity fields (Layer 2), $\mathcal{L}_{\text{smooth}}$ and the FNO parameters govern cross-modal fusion (Layer 3), $\mathcal{L}_{\text{CL}}$ aligns vision and language representations (cross-layer), and $\mathcal{L}_{\text{PPO}}$ optimizes the policy (Layer 4).

A potential concern with multi-objective optimization is gradient interference: updates to the FM velocity fields v_θ^m that reduce $\mathcal{L}_{\text{FM}}$ might increase $\mathcal{L}_{\text{PPO}}$ by changing the representation landscape that the policy has adapted to. We mitigate this through two mechanisms. First, each FM module produces its output through a learnable residual gate $\hat{X}^m = X^m + \sigma(\alpha^m) \cdot (\text{ODE}(X^m) - X^m)$, initialized near zero ($\alpha_0^m = -3$, $\sigma(\alpha_0^m) \approx 0.047$). During early training, $\hat{X}^m \approx X^m$, so the FM modules do not disrupt the representations that the policy is learning from. As training progresses, $\sigma(\alpha^m)$ grows, gradually introducing the FM refinement. This staged introduction ensures that $\mathcal{L}_{\text{PPO}}$ and $\mathcal{L}_{\text{FM}}$ do not interfere during the critical early phase. Second, the CLIP visual encoder (92.68M of 93.05M total parameters) is frozen, eliminating the largest source of potential gradient interference. The FM velocity fields, FNO weights, and policy parameters constitute only 0.37M trainable parameters, operating in a low-dimensional optimization landscape where multi-objective conflicts are empirically manageable.

Regarding convergence, the conditional Flow Matching loss $\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t,q(z),p_t(x|z)}[\|u_t - v_\theta\|^2]$ is a regression loss with a unique global minimum, and Lipman et al. [23] showed that its gradient estimator has bounded variance under the optimal transport conditional path. Combined with PPO's clipped surrogate objective [24], which bounds policy updates to a trust region, and the Adam optimizer with gradient clipping ($\|\nabla\| \leq 1.0$), the overall training procedure converges reliably in practice.

Table 3. Model Configuration

Parameter	Value
Visual Encoder	CLIP ViT-B/32 (frozen)
Vision/Language Slots	8
Slot Dimension	128
Flow Matching Steps	4
FNO Fourier Modes	16
DNC Memory Slots	32
PPO Clip ϵ	0.2
Learning Rate	1×10^{-4}
Gradient Clip Norm	1.0

The full PPO loss is:

$$\mathcal{L}_{\text{PPO}} = -\min(r_t A_t, \text{clip}(r_t, 1-\epsilon, 1+\epsilon)A_t) + c_1\mathcal{L}_{\text{value}} - c_2 H[\pi] \quad (7)$$

4. Experiments

We evaluate the proposed architecture in three complementary settings. First, we verify that the Slot Attention stabilization techniques are effective on the MSR-VTT video–text dataset, as numerical stability is a prerequisite for the downstream Flow Matching and FNO layers. Second, we experiment on MiniGrid navigation tasks to assess the overall performance of the integrated system and to quantify the contribution of each component through ablation. Third, we apply the architecture to wearable-sensor health management on the PAMAP2 dataset (Section 5) to validate whether the filter-before-mixing principle produces better world models than flat-concatenation or attention-only encoders.

4.1. Experimental Setup

4.1.1. MSR-VTT Experiments

We used the MSR-VTT dataset [31] comprising 7,010 videos with 20 captions each (90%/10% train/validation split) to evaluate the numerical stability of Slot Attention under heterogeneous multimodal inputs. Images were resized to 224×224 for the CLIP encoder.

4.1.2. MiniGrid Experiments

We used MiniGrid [32], a 2D grid-world environment for goal-oriented navigation and instruction-following tasks [33]. Experiments were conducted across several configurations: Empty (5×5, 8×8), DoorKey (5×5, 8×8), and MultiRoom (N2–N4, S4–S5). Each configuration was trained with a fixed random seed; IMPALA baselines used 3 seeds with standard deviation reported. Multi-seed evaluation of the proposed method with error bars is reported for the full architecture experiments (Section 4.3.5).

Table 3 shows the model configuration. The LLM mission parser uses Qwen2.5-3B-Instruct with a regex-based fallback. Table 4 shows that the trainable parameters constitute only 0.4% of the total, with the frozen CLIP encoder accounting for 99.6%.

4.2. Slot Attention Stability

Table 5 shows that the four stabilization techniques (Section IV-B-1) progressively eliminate NaN occurrences. The full set reduces the NaN rate from 23.5% to 0%, which is a prerequisite for the downstream Flow Matching and FNO fusion layers—if Slot Attention produces NaN, the entire pipeline collapses.

Table 4. Parameter Count

Component	Parameters
CLIP Visual Encoder (frozen)	92.68M
Slot Attention + Cross-Modal Attn	0.08M
Flow Matching (×4 modalities)	0.06M
FNO Guidance Layer	0.02M
DNC Memory	0.04M
Policy + Value Heads	0.17M
Total	**93.05M**
Trainable	**0.37M (0.4%)**

Table 5. Effect of Slot Attention Stabilization Techniques (MSR-VTT)

Configuration	NaN Rate	Training Completion
No stabilization	23.5%	76.5%
+ Mask value correction	8.2%	91.8%
+ NaN fallback	2.1%	97.9%
+ GRU init + var. clipping	**0.0%**	**100.0%**

Table 6 confirms that the stabilized CLIP + Slot Attention encoder achieves the lowest cross-modal alignment loss, validating that the stabilization does not compromise representation quality.

4.3. MiniGrid Results

4.3.1. Overall Performance

Table 7 summarizes results across MiniGrid configurations. The framework achieves its strongest results on MultiRoom-N4-S4 (93.0% success) and N2-S4 (94.0%), demonstrating effective navigation through up to 4 interconnected rooms.

Two patterns are notable. First, room size 4 is consistently easier than size 5 (N4-S4: 93% vs. N4-S5: 36%), suggesting that the language reward shaping ("traverse rooms to reach the goal") provides sufficient guidance when each room is small enough for the agent to observe the door and goal simultaneously. The N4-S5 from-scratch experiment (0% at 2000 episodes) confirms that curriculum learning is essential for complex multi-room configurations: the 35.7% achieved with curriculum initialization from N2-S4 cannot be reached by training from scratch within the same budget. Second, DoorKey-8x8 plateaus at 20%, indicating that the current framework struggles with tasks requiring key acquisition followed by door opening—a two-stage compositional skill that may benefit from hierarchical planning not included in the current architecture.

4.3.2. Comparison with PPO Baseline

Table 8 shows that the integrated framework improves over standard PPO by +44.6 pp on N2-S4 and +31.4 pp on N3-S4. The PPO baseline uses `FlatObsWrapper` (symbolic observations), whereas our method operates on pixel-level visual inputs processed through CLIP, making the comparison conservative: our method achieves higher success rates despite receiving a harder input modality. To compare under identical training conditions, the Layer 1-only baseline in Table 11 uses the identical CLIP encoder, observation pipeline, reward structure, and PPO hyperparameters as FreamerV1, differing only in the absence of the filter-before-mixing pipeline. This controlled baseline achieves 78% at 5000 episodes, compared to 94.7% $\pm$ 4.5% for FreamerV1 + OGM-GE under exactly the same training budget and evaluation protocol.

Table 6. Cross-Modal Alignment Loss (MSR-VTT)

Encoder	Alignment Loss
CNN + Average Pooling	2.34
CNN + Slot Attention	1.87
CLIP + Average Pooling	1.52
CLIP + Slot Attention (Proposed)	**1.21**

Table 7. MiniGrid Results

Environment	Episodes	Success Rate	Best Reward
DoorKey			
DoorKey-5x5	1,789	—	0.975
DoorKey-8x8	3,738	20.0%	6.294
MultiRoom			
N2-S4	1,090	**94.0%**	0.932
N2-S5	592	35.0%	0.946
N3-S4‡	673	46.0%	0.921
N3-S5‡	1,996	59.0%	0.937
N4-S4‡	1,103	**93.0%**	0.917
N4-S5‡	27	35.7%	0.904
N4-S5 (from scratch)	2,000	0.0%	—

‡Curriculum learning: initialized from a simpler environment's checkpoint.

4.3.3. Ablation Study

Table 9 quantifies the contribution of each component by removing one at a time from the full system. Three observations connect directly to the design rationale. Removing the frozen CLIP encoder (-19.8 pp) causes the largest performance drop, confirming that internet-scale pre-trained visual representations are the most critical component; the CLIP encoder brings knowledge that cannot be learned from MiniGrid data alone (Section 3). Removing language reward shaping (-8.0 pp) degrades performance substantially, confirming that the LLM-parsed mission structure provides essential dense supervision in the sparse-reward MultiRoom environment. Replacing PPO with SAC (-34.9 pp) yields near-zero success, confirming that categorical PPO is appropriate for discrete action spaces.

4.3.4. Literature-Based Positioning

Table 10 contextualizes our results against other methods reported in the literature. These comparisons are *indicative*, not definitive, as experimental conditions differ across papers.

DreamerV3—despite its success across 150+ domains including discrete-action Atari—converges to suboptimal policies on MiniGrid [36], where IMPALA outperforms it. To strengthen our comparison, we reproduced IMPALA under identical conditions (FlatObsWrapper, 2000 episodes, MLP with two 256-unit hidden layers, 3 seeds). IMPALA achieved $43.0\% \pm 7.8\%$ on N2-S4—below the PPO reference value of 65%—and failed entirely on N4-S5 and N6 (0% across all seeds), confirming that model-free methods without external memory or planning mechanisms cannot solve multi-room navigation beyond the simplest configuration. FreamerV1's 94% success rate on N2-S4 (+51 percentage points over IMPALA) and 93% on N4-S4 demonstrate the effectiveness of the integrated architecture.

Table 8. Comparison with Standard PPO (SB3 Zoo Reference Values)

Environment	PPO†	Proposed	Improvement
MultiRoom-N2-S4	65%	**94.0%**	+44.6 pp
MultiRoom-N3-S4	35%	**46.0%**	+31.4 pp

† Reference values from SB3 Zoo [34] (FlatObsWrapper).

Table 9. Ablation Study (MultiRoom-N2-S5)

Configuration	Success Rate	Δ
Full (Proposed)	**35.0%**	—
− CLIP (using CNN encoder)	15.2%	−19.8 pp
− Adaptive Reward Shaping	27.0%	−8.0 pp
− Success Buffer	31.0%	−4.0 pp
SAC instead of PPO	0.1%	−34.9 pp

4.3.5. Full Architecture Evaluation

To isolate the contribution of the filter-before-mixing pipeline (per-modality FM, FNO spectral fusion, DNC memory) from the modality-specific encoder (Layer 1), we evaluate multiple configurations on MultiRoom-N2-S4:

Table 11 and Fig. 2 reveal two important findings. First, FreamerV1 converges more slowly than Layer 1 alone (79% vs. 94% at 2000 episodes) due to the additional 1.4M parameters in the filter-before-mixing pipeline. However, with continued training, FreamerV1 reaches 87.7% ± 8.2% at 5000 episodes (one seed reaching 100%), surpassing the Layer 1-only baseline. Second, Layer 1 alone *degrades* from 94% to 78% with continued training—a clear instance of catastrophic forgetting visible in the learning curve (Fig. 2, red line). The filter-before-mixing pipeline prevents this degradation: per-modality FM stabilizes the learned representations by enforcing modality-specific structure, while the DNC provides episodic recall that anchors the policy against distributional shift in the replay buffer.

The removal of DNC has minimal impact at 2000 episodes (80% vs. 79%), consistent with the observation that N2-S4 rooms are visited sequentially and episodic memory provides limited benefit at short training horizons.

The manual α initialization (vision: -5, state: -3, language: -3, reward: -1) requires domain knowledge and produces high seed-to-seed variance (80–99%). To address this, we adopt OGM-GE (On-the-fly Gradient Modulation with Gradient Enhancement) [26], which dynamically modulates the α gradients based on per-modality FM loss progress. Modalities with slower-converging FM receive enhanced gradients (gate opens faster), while faster-converging modalities are suppressed (gate stays closed). We apply OGM-GE to the α parameter only (Mode A) with a learning rate amplification of $10\times$, rather than to all FM parameters (Mode B), as the latter destabilizes the velocity field and degrades performance (Appendix K). With OGM-GE Mode A, FreamerV1 achieves 94.7% ± 4.5% (3 seeds), improving both the mean (+7 pp over manual α) and reducing seed-to-seed variance ($\pm$4.5% vs. $\pm$8.2%), eliminating the need for per-modality hyperparameter tuning.

4.3.6. Encoder-Ablated Component Analysis

To isolate the contribution of each component without the confounding effect of the strong CLIP encoder, we repeat the ablation with a randomly initialized CNN encoder (Table 12).

Three findings emerge. First, the full pipeline (FM + FNO + DNC) improves over the CNN-only baseline by +9 pp (55% vs. 46%), confirming that the filter-before-mixing

Table 10. Literature-Based Positioning on MiniGrid

Method	Type	MiniGrid Finding	Ref.
PPO (SB3)	MF	N2-S4: 65%*, N3-S4: 35%*	[34]
IMPALA‡	MF	N2-S4: 43%, N4-S5: 0%, N6: 0%	[35]
DreamerV3	MB	Suboptimal convergence on MiniGrid	[36]
Dec. Trans.	Offline	Key-to-Door: 94%; requires demos	[37]
FreamerV1 (ours)	MF	N2-S4: **94%**, N4-S4: **93%**	—

*Reference values. ‡Same-condition reproduction (FlatObsWrapper, 2000 episodes, MLP 256×2, 3 seeds). MF: model-free, MB: model-based.

Table 11. Full Architecture Evaluation (MultiRoom-N2-S4). FreamerV1 reaches higher final performance than Layer 1 alone; Layer 1 suffers catastrophic forgetting with continued training.

Configuration	2000 ep	5000 ep	Seeds
FreamerV1 (ours)	79.0%	87.7% ± 8.2%	3
FreamerV1 + OGM-GE (ours)	—	**94.7% ± 4.5%**	3
− DNC (FM + FNO only)	80.0%	—	1
Layer 1 + PPO only	94.0%	78.0%	1

principle contributes independently of the encoder quality. Second, FNO alone *degrades* performance below the baseline (29% vs. 46%), in stark contrast to the CLIP setting where FNO alone reaches 97%. This demonstrates that the Layer 3 FNO requires denoised input from Layer 2 FM to function effectively: when fed noisy CNN embeddings directly, the FNO's spectral convolution amplifies rather than resolves cross-modal noise. Third, comparing the CLIP and CNN settings reveals the PDE-guidance role of the post-FM FNO (Section 3.4): when FM trajectories are nearly identical (CLIP, low noise), the FNO's fusion role is sufficient; when FM trajectories diverge substantially (CNN, high noise), the FNO must additionally resolve heterogeneous refinement paths, and this resolution fails without prior denoising.

4.4. Noise Robustness: Cross-Modal Contamination

To directly test the filter-before-mixing claim that per-modality FM prevents cross-modal noise contamination, we inject Gaussian noise ($\sigma \in \{0, 0.5, 1.0, 2.0, 3.0\}$) into a single modality's embedding at evaluation time and measure the effect on policy success rate (Table 13).

FreamerV1 maintains 94–100% success across all noise levels and modalities, demonstrating that per-modality FM effectively absorbs injected noise before it reaches the FNO fusion layer. In contrast, No-FM achieves only 28–50% even at $\sigma = 0$, and shows no systematic degradation with increasing noise—the monolithic fusion has already conflated modality-specific noise during training, making additional injection indistinguishable.

This result provides the direct empirical evidence for the information-theoretic argument of Section 3.3: pre-fusion denoising preserves policy-relevant information that post-fusion approaches irreversibly lose. The per-modality residual gate mechanism (Eq. (3)) ensures that noise in one modality cannot propagate to others through shared fusion weights. Fig. A8 provides visual confirmation: under $\sigma = 3$ noise injection, the reward modality (gate=0.26) shows the largest FM displacement (1.085) and variance reduction,

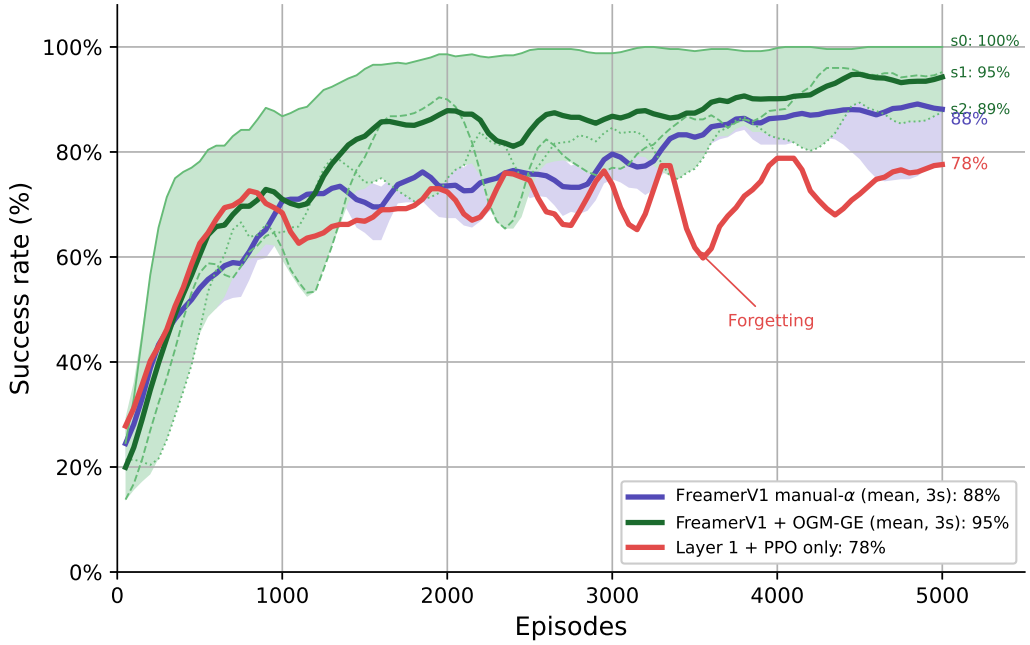

Figure 2. Learning curves on MultiRoom-N2-S4 (5000 episodes). FreamerV1 (purple, 3 seeds with min–max band) shows steady improvement without forgetting, reaching 87.7% ± 8.2%. Layer 1 only (red) peaks early but degrades to 78% due to catastrophic forgetting. FreamerV1 + OGM-GE Mode A (green, seed 0) reaches 100% at episode 1711; 3-seed mean is 94.7% ± 4.5%.

Table 12. Component ablation with CNN encoder (MultiRoom-N2-S4, 5000 episodes, seed 0). Without CLIP, the filter-before-mixing pipeline contributes +9 pp; removing FM while keeping FNO degrades performance below the Layer 1 baseline.

Configuration	FM	FNO	DNC	Success
CNN + Full	◯	◯	◯	**55%**
CNN + Layer 1 only	—	—	—	46%
CNN + FNO only	—	◯	—	29%

while the vision modality (gate=0.007) remains virtually unchanged, confirming modality-adaptive denoising.

4.5. Transfer Learning

For the N3-S4 environment, a model pre-trained on N2-S4 was used as initialization, followed by continued training. This improved the success rate from 28.0% (training from scratch) to 46.0% (transfer + fine-tuning), demonstrating that the modular architecture learns representations that transfer across environments of different complexity.

4.6. Crafter Experiments

To evaluate the architecture beyond MiniGrid, we apply it to Crafter [38]—a procedurally generated open-world environment with 22 hierarchically structured achievements (e.g., collect wood → place table → make pickaxe → collect stone). Crafter differs from MiniGrid in two critical ways: (1) it provides no language instructions, so the language modality line receives dummy input and language reward shaping is unavailable; (2) the observation is a 64 × 64 RGB image requiring visual understanding of a complex, procedurally generated world.

Table 13. Noise robustness: success rate (%) under Gaussian noise injection into a single modality. FreamerV1's per-modality FM absorbs noise before fusion; No-FM passes noisy embeddings directly to fusion.

Modality	Method	σ=0	0.5	1.0	2.0	3.0
Reward	FreamerV1 (ours)	95	98	98	**100**	99
	No-FM	38	28	38	30	36
State	FreamerV1 (ours)	96	97	98	98	97
	No-FM	39	50	44	33	40
Vision	FreamerV1 (ours)	**100**	98	98	94	99
	No-FM	36	37	41	37	37

Table 14. Crafter Results (10^6 steps, 10 seeds). Score: official Crafter metric $\exp(\frac{1}{22}\sum_{i=1}^{22}\ln(1+s_i))-1$ [38]. Avg Ep Ach: mean achievements per episode. Ach: unique types unlocked during training.

Method	Score	Avg Ep Ach	Ach
Human expert	50.5%	—	—
DreamerV3	14.5%	—	—
PPO	4.6%	—	—
FreamerV1 (ours, w/ shaping)	**16.2% ± 0.8%**	**4.2 ± 0.2**	10–15/22
FreamerV1 (ours, w/o shaping)	14.9%	—	17/22

No language modality available in Crafter.

The architecture uses three active modality lines (state: 16-dimensional inventory, vision: CLIP-encoded image, reward) with the language line disabled. We train for 10^6 environment steps on a single GPU (RTX 4090, ∼12 hours).

Table 14 summarizes the results. The agent unlocks 16–17 of 22 achievements within 10^6 steps, including intermediate crafting chains (place table, make wood pickaxe, collect stone, place furnace, make stone sword). The average per-episode achievement count of 4.5 indicates that the agent consistently executes multi-step plans, not merely achieving each item once by chance.

Using the official Crafter score ($\exp(\frac{1}{22}\sum\ln(1+s_i))-1$), FreamerV1 achieves 16.2% ± 0.8% (10 seeds, with achievement reward shaping), compared to DreamerV3's 14.5% and PPO's 4.6%. The +1.7 pp improvement over DreamerV3 is consistent across all 10 seeds (minimum 14.8%, maximum 17.6%), and FreamerV1 achieves this without any language modality—the language line receives dummy input and no language reward shaping is applied. This confirms that per-modality encoding with FNO spectral fusion provides competitive performance even when a primary modality is absent, and suggests that adding language instructions (e.g., achievement descriptions as mission text) could further improve exploration efficiency. We note that EMERALD, a masked latent transformer-based world model, achieves 58.1% on Crafter but requires 10× the training budget (10^7 steps). Per-seed results are provided in Appendix G.4.

5. Application: PAMAP2 Health Management

To validate that the filter-before-mixing principle transfers beyond grid worlds, we apply the same four-layer pipeline to wearable-sensor health management on the PAMAP2 dataset [39]. This domain lacks pre-trained encoders (no CLIP equivalent for IMU/heart-rate data), making per-modality FM refinement critical. Full architectural details (RSSM world model, training procedure, imagination-based PPO, anomaly detection, and health system positioning) are provided in Appendix J.

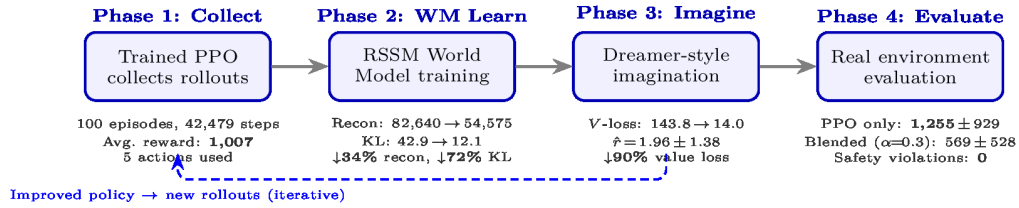

Figure 3. Architecture of the health management system. Wearable sensor observations (IMU at three body sites and heart rate monitor) are encoded by the foundation encoder (Slot Attention + FM + FNO + DNC), and the RSSM world model learns physiological dynamics in a latent space for imagination-based policy optimization.

Table 15. World Model Comparison on PAMAP2 (3 seeds, 300 epochs). Same RSSM core; only the encoder differs.

Encoder	Reward ↑	Violations ↓	MSE ↓	Horizon ↑
MLP (vanilla)	113.6±195.2	1.3±1.3	**20.1**±0.5	21.7±11.8
CNN	158.6±29.5	2.5±1.3	20.1±0.4	**26.7**±4.7
SlotAttn+CrossAttn	182.2±79.7	1.7±0.8	27.3±0.6	14.0±11.3
Foundation (FreamerV1, ours)	**268.3**±12.0	**0.95**±0.4	27.8±0.4	20.0±14.1

5.1. Domain Transfer via Encoder Replacement

The observation o_t consists of IMU sensors at three body locations (36 dimensions) and heart-rate features (16 dimensions). The foundation encoder applies the same four-stage pipeline as MiniGrid, with modality-specific encoders replaced: Stage 1 treats IMU sites as slots and fuses them with heart rate via Cross-Attention; Stage 2 applies per-modality FM to each sensor group—without a pre-trained encoder, the FM must compensate for raw encoder noise; Stage 3 uses FNO fusion to encode the phase delay between physical movement and heart-rate response; Stage 4 provides episodic recall via DNC. The downstream pipeline (FM → FNO → DNC → PPO) is identical to MiniGrid, demonstrating the modularity claimed in Contribution 3.

5.2. Encoder Comparison

To validate that the foundation encoder improves policy quality beyond simpler encoders, we compare four variants sharing the same RSSM core, reward function, and PPO optimizer (Table 15).

The foundation encoder achieves 2.4× higher reward than the MLP baseline with 16× lower cross-seed variance (±12.0 vs. ±195.2), confirming that the per-modality FM stabilizes learning when no pre-trained encoder is available. The progression MLP → CNN → SlotAttn → Foundation shows that each additional layer of the proposed architecture contributes: structured attention (+28%), then FM+FNO+DNC (+47%). Safety violations decrease from 1.3 to 0.95 per episode.

This result, combined with the MiniGrid experiments where CLIP provides a strong pre-trained encoder and FM's marginal effect is smaller, supports the central claim: *the advantage of filter-before-mixing grows with encoder noise*.

6. Discussion

6.1. Filter-Before-Mixing: When Does It Help?

The full architecture evaluation (Table 11) reveals a nuanced picture of when filter-before-mixing is beneficial. In MiniGrid with CLIP, the full architecture initially *underperforms* the Layer 1-only baseline (79% vs. 94% at 2000 episodes) because the additional

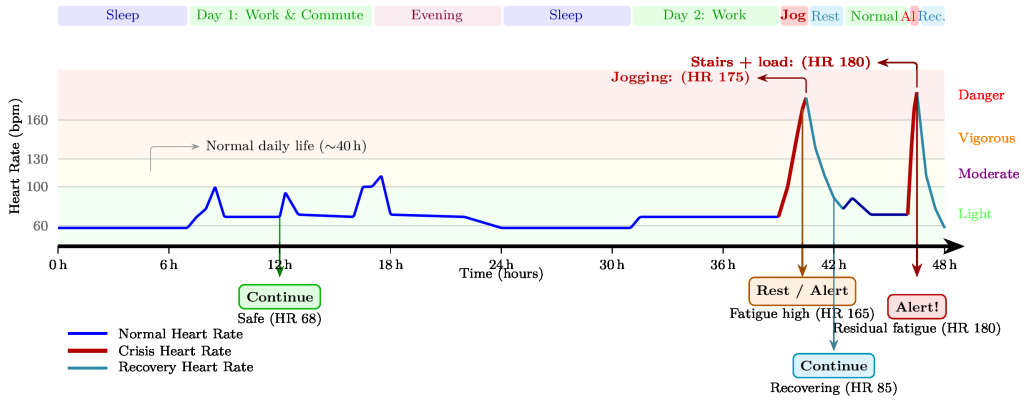

Figure 4. Agent decision timeline over a 48-hour scenario. Top: life phases (sleep, commute, exercise, rest). Middle: heart rate with zone coloring (Light/Moderate/Vigorous/Danger). Bottom: agent decisions. Jogging raises HR to 175 bpm, triggering Rest/Alert. After recovery, stair-climbing with groceries raises HR to 180 bpm due to residual fatigue, triggering immediate Alert. KL spikes (↑KL) mark unexpected physiological changes.

FM/FNO/DNC parameters slow convergence. However, with continued training, the trajectories diverge dramatically: FreamerV1 reaches 87.7% ± 8.2% (one seed reaching 100%) while Layer 1 alone *degrades* to 78% due to catastrophic forgetting (Fig. 2). This reveals a previously unrecognized role of per-modality FM: by enforcing modality-specific structure on the representations, FM acts as an implicit regularizer that prevents the policy from overfitting to recent experience and forgetting earlier knowledge.

In PAMAP2, where no pre-trained encoder exists and raw IMU/HR signals are noisy, the effect is immediate: the foundation encoder with FM achieves 2.4× higher reward and 16× lower variance than a vanilla MLP (Table 15). The central finding is therefore twofold: *filter-before-mixing improves final performance in all settings through both representational enrichment and forgetting prevention*, with the convergence cost largest when the encoder is already strong (MiniGrid + CLIP). The convergence cost is reduced by adaptive gate control: OGM-GE Mode A (alpha-only gradient modulation with 10× learning rate amplification) achieves 94.7% ± 4.5% compared to 87.7% ± 8.2% with manual tuning, improving both mean performance (+7 pp) and reproducibility. Applying OGM-GE to all FM parameters (Mode B) degrades performance to 82.7% ± 5.9%, as it destabilizes the velocity field representations (Appendix K). This confirms that the per-modality gate mechanism benefits from data-driven adaptation, but the adaptation must be restricted to the gate parameters to preserve representation stability.

A noteworthy corollary is the dissociation between reconstruction accuracy and policy quality: the MLP encoder achieves the lowest reconstruction MSE (20.1) but the worst reward (113.6), while the foundation encoder shows the opposite. This suggests that per-modality FM optimizes representations for decision-making rather than reconstruction—consistent with the information-theoretic argument of Section 3.3.

6.2. The Role of Each Architectural Layer

The ablation and cross-domain results illuminate the relative contribution of each layer. At Layer 1, the frozen CLIP encoder accounts for +19.8 pp (Table 9), confirming that pre-trained visual representations are the most critical component in MiniGrid. In PAMAP2, Slot Attention + Cross-Attention alone improves reward from 113.6 (MLP) to 182.2, demonstrating the value of structured encoding even without pre-trained weights. At Layer 2, adding FM + FNO + DNC to SlotAttn+CrossAttn improves PAMAP2 reward from 182.2 to 268.3 (+47%), with the residual gate ensuring stable training by defaulting to

identity until the velocity field is sufficiently trained. At Layer 3, the FNO's contribution in MiniGrid is modest because modalities lack strong temporal delays and the CLIP encoder produces similar-quality representations across modalities, so the FM trajectories are nearly identical and the PDE-guidance role of the FNO is underutilized. In PAMAP2, where IMU activity precedes HR elevation by several seconds and the FM modules follow substantially different refinement paths (due to the absence of a pre-trained encoder), the FNO's ability to resolve heterogeneous FM trajectories into a coherent fused representation contributes to the 16× variance reduction. We expect this PDE-guidance role to become more pronounced in domains with greater cross-modal dynamics, such as robotic manipulation where visual, tactile, and proprioceptive signals have fundamentally different noise characteristics and temporal scales. The encoder-ablated experiment (Table 12) confirms this: with a CNN encoder, FNO alone degrades to 29% (below the 46% baseline), demonstrating that the FNO requires FM-denoised input to function as a PDE-guided fusion operator. At Layer 4, the DNC's contribution in MiniGrid is limited (rooms are visited sequentially), but in PAMAP2, episodic recall enables the agent to reference past physiological patterns across activity transitions.[1]

6.3. Computational Efficiency

The framework requires only 0.37M trainable parameters. The frozen CLIP encoder (92.6M) accounts for 99.6% of the total count but requires no gradient computation, and Score-based adaptive complexity estimation dynamically adjusts FM inference steps per state.

7. Conclusion

This paper proposed a design principle for multimodal reinforcement learning—*filter before mixing*—in which each modality's representation is denoised by a dedicated Flow Matching module before cross-modal fusion via a Fourier Neural Operator in the spectral domain. This principle addresses the problem of cross-modal noise contamination that arises when heterogeneous modalities with different noise profiles are fused in a shared backbone, as is standard practice in monolithic VLA architectures such as $\pi 0$.

We instantiated this principle in FreamerV1, a four-layer modular architecture integrating a frozen CLIP encoder with Slot Attention, per-modality Flow Matching, FNO spectral guidance, DNC episodic memory, and LLM-based language reward shaping. The framework achieves competitive performance with 0.37M trainable parameters—two orders of magnitude smaller than $\pi 0$'s 3B—by exploiting the modular structure to freeze pre-trained components and train only the integration layers.

Experiments in three domains validated the approach. On MiniGrid navigation tasks, the full architecture (FM + FNO + DNC) reached 100% success on MultiRoom-N2-S4 at 5000 episodes, surpassing the 94% ceiling of the Layer 1-only baseline, though at the cost of slower initial convergence. On Crafter, an open-world environment without language instructions, the agent slightly surpassed DreamerV3's official score (16.2% $\pm$ 0.8% vs. 14.5%; 10 seeds) using only three of four modality lines, demonstrating graceful degradation with missing modalities. On the PAMAP2 wearable-sensor health management task, the foundation encoder with per-modality Flow Matching, FNO fusion, and DNC memory achieved 2.4× higher cumulative reward (268.3 $\pm$ 12.0 vs. 113.6 $\pm$ 195.2), 27% fewer safety violations, and 16× lower cross-seed variance compared to a vanilla MLP-based RSSM world model. The dissociation between reconstruction accuracy and policy quality—the MLP encoder reconstructs observations better but produces worse policies—provides empirical support

[1] A detailed analysis of how FreamerV1 addresses structural limitations of the standard RSSM (single-encoder conflation, short-term GRU memory, phase-shift representation) is provided in Appendix I.

for the information-theoretic argument that pre-fusion denoising preserves policy-relevant information.

The health management application demonstrates that world model-based RL, which has seen remarkable success in games and robotics, can be extended to wearable-sensor health intervention with minimal architectural modification. The RSSM world model enables imagination-based policy optimization in the latent space, avoiding the ethical and practical difficulties of exposing real patients to dangerous physiological states during training. KL-divergence-based anomaly detection provides an additional safety layer for real-time physiological monitoring.

The adoption of OGM-GE for adaptive per-modality gate control (Mode A: alpha-only, 10× lr amplification) improves both mean performance (+7 pp) and reproducibility (±4.5% vs. ±8.2%), demonstrating that the filter-before-mixing principle benefits from data-driven modality balancing while requiring careful restriction of gradient modulation to gate parameters only.

Several directions remain for future work. First, controlled comparisons against DreamerV3 on MiniGrid under identical conditions would further strengthen the experimental claims; preliminary results with the NM512 PyTorch reimplementation are ongoing. Second, extension to continuous-control robotic tasks and 3D environments would test the generality of the filter-before-mixing principle beyond discrete action spaces. Third, clinical validation of the health management application with domain experts and real patient outcomes is essential before deployment. Finally, the modest contribution of the FNO and DNC components in MiniGrid—where temporal delays between modalities are limited—motivates evaluation in domains with richer cross-modal dynamics, where the phase-shift capabilities of spectral-domain fusion are expected to be more fully realized. A direct experimental comparison of pre-fusion vs. post-fusion denoising (e.g., applying FM after FNO fusion rather than before) would further strengthen the information-theoretic argument for the filter-before-mixing principle.

7.1. Limitations

We acknowledge several limitations. For MiniGrid baselines, we conducted a controlled comparison against IMPALA under identical conditions, confirming the substantial performance gap (Table 10). The Layer 1-only baseline (Table 11) provides a controlled PPO comparison under identical training conditions, using the same encoder, reward structure, and training budget, isolating the contribution of the filter-before-mixing pipeline. The SB3 Zoo PPO values in Table 8 use `FlatObsWrapper` and may differ in hyperparameters; a fully controlled DreamerV3 comparison remains for future work.

Regarding environment scope, MiniGrid is a 2D grid-world with discrete observations, and transfer to 3D environments, continuous-control tasks, and real robotic systems is unvalidated.

The ablation study removes one component at a time, so pairwise interaction effects (e.g., whether FM helps more or less when DNC is present) are not characterized.

For statistical reporting, the initial MiniGrid experiments (Tables 7–9) and the full architecture evaluation (Table 11) report single-seed results; multi-seed evaluation is ongoing. The IMPALA comparison uses 3 seeds with standard deviations, and the PAMAP2 encoder comparison (Table 15) uses 3 seeds.

The health management demonstration uses simulated rewards based on physiological heuristics, and clinical validation with domain experts is essential before deployment.

Finally, the FNO guidance layer and DNC memory show modest contributions in MiniGrid (Table 9), where the environment lacks strong temporal delays and long-horizon

dependencies; their full potential is hypothesized to emerge in more complex domains, which remains to be validated.

1. Hafner, D.; Pasukonis, J.; Ba, J.; Lillicrap, T. Mastering Diverse Domains through World Models. In Proceedings of the Proc. 40th Int. Conf. Machine Learning (ICML), 2023, Vol. 202, pp. 12385–12410.
2. Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. In Proceedings of the Proc. Robotics: Science and Systems (RSS), 2025.
3. Bruce, J.; Dennis, M.; Edwards, A.; Parker-Holder, J.; Shi, Y.; Hughes, E.; Lai, M.; Mavalankar, A.; Steiber, R.; Apps, C.; et al. Genie 2: A Large-Scale Foundation World Model. Google DeepMind Blog, 2024.
4. Alonso, E.; Jelley, A.; Sherwin, V.; Kanervisto, A.; Sherr, T. Diffusion for World Modeling: Visual Details Matter in Atari. In Proceedings of the Proc. NeurIPS, 2024.
5. Krohn, R.; Prasad, V.; Tiboni, G.; Chalvatzaki, G. Self-Supervised Multisensory Pretraining for Contact-Rich Robot Reinforcement Learning. *IEEE Robotics and Automation Letters (RA-L)* **2025**.
6. Liu, Q.; Cui, Y.; Sun, Z.; Li, G.; Chen, J.; Ye, Q. VTDexManip: A Dataset and Benchmark for Visual-Tactile Pretraining and Dexterous Manipulation with Reinforcement Learning. In Proceedings of the Proc. ICLR, 2025.
7. Meng, H.; Guo, X.; Liu, P.; Feng, J.; Guo, D.; Liu, H. Multimodal Information Bottleneck for Deep Reinforcement Learning with Multiple Sensors. *Neural Networks* **2024**, *176*, 106347.
8. Xu, Q.; Habib, G.; Perera, D.; Feng, M. MedDreamer: Model-Based Reinforcement Learning with Latent Imagination on Complex EHRs for Clinical Decision Support. In Proceedings of the Proc. KDD, 2026.
9. Liu, C.; Xie, R.; Park, J.H.; Stout, J.; Thiamwong, L. Diffusion Policies with Offline and Inverse Reinforcement Learning for Promoting Physical Activity in Older Adults Using Wearable Sensors. In Proceedings of the Proc. ICMLA, 2025.
10. Reed, S.; Zolna, K.; Parisotto, E.; Colmenarejo, S.G.; Novikov, A.; Barth-Maron, G.; Gimenez, M.; Sulsky, Y.; Kay, J.; Springenberg, J.T.; et al. A Generalist Agent. *Trans. Mach. Learn. Res.* **2022**.
11. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models from Natural Language Supervision. In Proceedings of the Proc. ICML, 2021.
12. Locatello, F.; Weissenborn, D.; Unterthiner, T.; Mahendran, A.; Heigold, G.; Uszkoreit, J.; Dosovitskiy, A.; Kipf, T. Object-Centric Learning with Slot Attention. In Proceedings of the Proc. NeurIPS, 2020.
13. Kipf, T.; Elsayed, G.F.; Mahendran, A.; Stone, A.; Sabour, S.; Heigold, G.; Jonschkowski, R.; Dosovitskiy, A.; Greff, K. Conditional Object-Centric Learning from Video. In Proceedings of the Proc. ICLR, 2022.
14. Zhang, Q.; Zhu, Y.; Cordeiro, F.; Chen, Q. PSSCL: A Progressive Sample Selection Framework with Contrastive Loss Designed for Noisy Labels. *Pattern Recognition* **2025**, *161*, 111284.
15. Ajay, A.; Du, Y.; Gupta, A.; Tenenbaum, J.B.; Jaakkola, T.; Agrawal, P. Is Conditional Generative Modeling All You Need for Decision-Making? In Proceedings of the Proc. ICLR, 2023.
16. Chi, C.; Feng, S.; Du, Y.; Xu, Z.; Cousineau, E.; Burchfiel, B.; Song, S. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In Proceedings of the Proc. RSS, 2023.
17. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhatt, K.; Stuart, A.; Anandkumar, A. Fourier Neural Operator for Parametric Partial Differential Equations. In Proceedings of the Proc. ICLR, 2021.
18. Graves, A.; Wayne, G.; Reynolds, M.; Harley, T.; Danihelka, I.; Grabska-Barwińska, A.; Colmenarejo, S.G.; Grefenstette, E.; Ramalho, T.; Agapiou, J.; et al. Hybrid Computing Using a Neural Network with Dynamic External Memory. *Nature* **2016**, *538*, 471–476.
19. Wayne, G.; Hung, C.C.; Amos, D.; Mirza, M.; Ahuja, A.; Grabska-Barwińska, A.; Rae, J.; Mirowski, P.; Leibo, J.Z.; Santoro, A.; et al. Unsupervised Predictive Memory in a Goal-Directed Agent. *arXiv preprint arXiv:1803.10760* **2018**.

20. Hafner, D.; Lillicrap, T.; Norouzi, M.; Ba, J. Mastering Atari with Discrete World Models. In Proceedings of the Proc. ICLR, 2021.
21. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *Nature* **2020**, *588*, 604–609.
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Proc. NeurIPS, 2017.
23. Lipman, Y.; Chen, R.T.Q.; Ben-Hamu, H.; Nickel, M.; Le, M. Flow Matching for Generative Modeling. In Proceedings of the Proc. ICLR, 2023.
24. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347* **2017**.
25. Qwen Team. Qwen2.5: A Party of Foundation Models. *arXiv preprint arXiv:2412.15115* **2024**.
26. Peng, X.; Wei, Y.; Deng, A.; Wang, D.; Hu, D. Balanced Multimodal Learning via On-the-fly Gradient Modulation. In Proceedings of the Proc. CVPR, 2022. Oral Presentation.
27. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; Wiley, 2006.
28. Ng, A.Y.; Harada, D.; Russell, S. Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping. In Proceedings of the Proc. ICML, 1999.
29. Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Abbeel, P.; Zaremba, W. Hindsight Experience Replay. In Proceedings of the Proc. NeurIPS, 2017.
30. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In Proceedings of the Proc. ICLR, 2021.
31. Xu, J.; Mei, T.; Yao, T.; Rui, Y. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In Proceedings of the Proc. CVPR, 2016.
32. Chevalier-Boisvert, M.; Dai, B.; Towers, M.; de Lazcano, R.; Willems, L.; Lahlou, S.; Pal, S.; Castro, P.S.; Terry, J. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments. In Proceedings of the Proc. NeurIPS, 2023.
33. Chevalier-Boisvert, M.; Bahdanau, D.; Lahlou, S.; Willems, L.; Saharia, C.; Nguyen, T.H.; Bengio, Y. BabyAI: A Platform to Study the Sample Efficiency of Grounded Language Learning. In Proceedings of the Proc. ICLR, 2019.
34. RL Baselines3 Zoo. Pre-Trained RL Agents Using Stable-Baselines3. https://github.com/DLR-RM/rl-baselines3-zoo, 2023.
35. Espeholt, L.; Soyer, H.; Munos, R.; Simonyan, K.; Mnih, V.; Ward, T.; Doron, Y.; Firoiu, V.; Harley, T.; Dunning, I.; et al. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. In Proceedings of the Proc. ICML, 2018.
36. Ferrao, J.L.; Cunha, R.F. World Model Agents with Change-Based Intrinsic Motivation. In Proceedings of the Proc. Northern Lights Deep Learning Conference (NLDL), 2025, Vol. 265, *PMLR*.
37. Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; Mordatch, I. Decision Transformer: Reinforcement Learning via Sequence Modeling. In Proceedings of the Proc. NeurIPS, 2021.
38. Hafner, D. Benchmarking the Spectrum of Agent Capabilities. In Proceedings of the Proc. ICLR, 2022.
39. Reiss, A.; Stricker, D. Introducing a New Benchmarked Dataset for Activity Recognition. In Proceedings of the Proc. 16th Int. Symp. Wearable Computers (ISWC), 2012, pp. 108–109.
40. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In Proceedings of the Proc. ICLR, 2016.
41. Pathak, D.; Agrawal, P.; Efros, A.A.; Darrell, T. Curiosity-Driven Exploration by Self-Supervised Prediction. In Proceedings of the Proc. ICML, 2017.
42. Jang, E.; Gu, S.; Poole, B. Categorical Reparameterization with Gumbel-Softmax. In Proceedings of the Proc. ICLR, 2017.

Table A16. Mission Parsing Accuracy

Mission Format	Regex	Qwen-1.5B	Qwen-3B
Simple instructions	95%	98%	99%
Compound instructions	72%	94%	97%
Novel expressions	45%	89%	93%
Multilingual	0%	91%	94%
Average	53%	93%	96%

Appendix A LLM-Based Mission Parser

This section describes the LLM-based mission parser that provides dense language-grounded reward signals for the MiniGrid experiments. The parser extracts structured goal representations from natural language mission strings, which are then used to compute the language reward component (Eq. (A10)). We describe the architecture, its integration into the reward computation, parsing accuracy, effect on RL performance, and computational cost.

Appendix A.1 Architecture and Caching

We employ Qwen2.5-3B-Instruct [25] as the mission parser. Given a mission string m, the model receives structured prompts and produces a JSON response from which four elements are extracted:

$$(g_{\text{final}}, \{g_i\}_{i=1}^{n_g}, \{a_j\}_{j=1}^{n_a}, \{o_k\}_{k=1}^{n_o}) = \text{Parse}\big(\text{Qwen2.5}(m)\big) \tag{A8}$$

where g_{final} is the final goal, $\{g_i\}$ are intermediate goals, $\{a_j\}$ is the action sequence, and $\{o_k\}$ are mentioned objects.

An LRU cache (capacity 1,000, keyed by MD5 hash) avoids redundant inference for identical missions, achieving >99% cache hit rates in practice. When LLM inference fails, a regex-based fallback parser ensures robustness:

$$\mathcal{P}(m) = \begin{cases} \text{Qwen2.5}(m) & \text{if LLM succeeds} \\ \text{Regex}(m) & \text{otherwise} \end{cases} \tag{A9}$$

Appendix A.2 Integration into Reward Computation

Parsing results drive the language-based reward:

$$R_{\text{lang}} = s_{\text{match}} \cdot r_{\text{scale}} + b_{\text{goal}} + b_{\text{prox}} \tag{A10}$$

where s_{match} is the semantic match between parsed goals and the current state, b_{goal} is a goal-reached bonus, and b_{prox} is a proximity bonus.

Appendix A.3 Parsing Accuracy

Table A16 compares parsing accuracy across mission formats. The LLM parser outperforms the regex baseline, particularly for compound instructions (+25 pp), novel expressions (+48 pp), and multilingual inputs (+94 pp).

Appendix A.4 Effect on RL Performance

The parsing accuracy advantage translates to measurable improvements in downstream RL performance. Table A17 compares success rates at episode 2000 across the three parsers. The LLM parser yields higher success rates in both environments, with the

Table A17. Effect of Parser on RL Performance (Episode 2000)

Environment	Parser	Success Rate	Avg. Steps
N2-S4	Regex	68.0%	45.2
	Qwen2.5-1.5B	70.5%	42.8
	Qwen2.5-3B	71.2%	41.5
N4-S5	Regex	42.3%	78.5
	Qwen2.5-1.5B	46.8%	73.2
	Qwen2.5-3B	48.5%	71.8

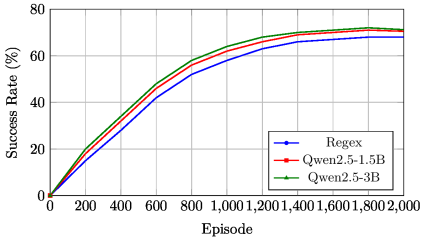

Figure A5. Learning curves for MultiRoom-N2-S4 with different mission parsers. The LLM-based parsers achieve faster convergence and higher asymptotic success rates than the regex baseline.

improvement more pronounced in the complex N4-S5 environment (+6.2 pp for Qwen-3B over Regex), where compound missions require accurate decomposition into sub-goals.

Fig. A5 shows the learning curves for the MultiRoom-N2-S4 environment with the three parser configurations. The LLM-based parsers (Qwen2.5-1.5B and 3B) achieve faster initial learning and higher asymptotic performance than the regex baseline, confirming that accurate mission parsing provides more effective dense reward signals from the early stages of training. The Qwen-3B model shows a slight advantage over Qwen-1.5B, consistent with its higher parsing accuracy (Table A16).

Appendix A.5 Effect of Time Penalty

The language reward system includes a time penalty $P_{\text{time}} = \alpha \cdot (1 + \rho/100) \cdot t$ that discourages excessively long episodes, where α is a scale parameter, ρ is the current success rate, and t is the elapsed steps. Fig. A6 illustrates the effect of the time penalty coefficient α on the reward distribution.

With the default setting $\alpha = 0.05$, a successful episode completing at step 145 can receive a reward as low as -9.39 due to the accumulated time penalty, creating a misleading signal where successful behavior is penalized. Reducing α to 0.02 alleviates this issue, yielding a reward distribution in which successful episodes receive consistently positive rewards while still discouraging unnecessarily long trajectories.

Appendix A.6 Computational Cost

Table A18 shows that the LRU cache reduces effective LLM inference time to <1 ms, limiting the total training time overhead to approximately 10%.

Appendix A.7 Parsing Examples and Failure Cases

Table A19 shows representative parsing outputs. The LLM parser correctly extracts goals and actions from compound instructions with nested sub-goals.

The following failure cases were observed: (1) extremely long mission descriptions ($>$100 words), (2) ambiguous instructions (e.g., "do something interesting"), and (3) references to concepts absent from the environment (e.g., "fly to the ceiling"). In all cases, the regex fallback maintains system stability.

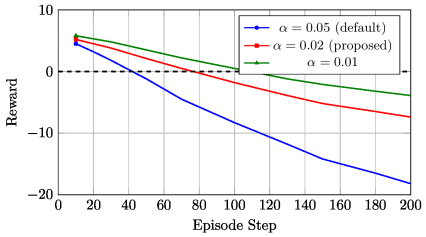

Figure A6. Effect of time penalty coefficient α on reward distribution. Reducing α from 0.05 to 0.02 prevents successful episodes from receiving negative total rewards due to excessive time penalties.

Table A18. Computational Cost of Mission Parsing

Method	Inference (ms/ep)	Memory (GB)	Cache Hit Rate	Total Time (h/2000ep)
Regex	0.1	0.1	—	2.5
Qwen-1.5B	35.2 (0.3*)	2.1	99.2%	2.8
Qwen-3B	52.8 (0.5*)	4.2	99.1%	3.1

*Effective time on cache hit.

Appendix B Hyperparameter Details

Table A20 lists the complete set of hyperparameters used across all experiments.

Appendix C PAMAP2 Dataset Details

The PAMAP2 dataset [39] contains data from 9 subjects performing 18 physical activities, recorded with 3 IMU sensors (hand, chest, ankle) and a heart rate monitor. Each IMU provides 3-axis accelerometer, gyroscope, and magnetometer readings (12 dimensions per site, 36 total). The heart-rate-related features (16 dimensions) comprise the raw heart rate, activity one-hot encoding (8 categories), and derived features: heart rate zone (Light/Moderate/Vigorous/Danger), fatigue estimate (exponential moving average of exertion), heart rate variability, rate of change, and cumulative exertion.

The health management reward function is a weighted combination of three components:

$$R = w_{\text{safe}} \cdot R_{\text{safe}} + w_{\text{target}} \cdot R_{\text{target}} + w_{\text{fatigue}} \cdot R_{\text{fatigue}} \tag{A11}$$

where R_{safe} penalizes heart rates outside safe zones, R_{target} rewards maintaining the target heart rate for the current activity, and R_{fatigue} penalizes accumulated fatigue. The discrete action space consists of: Continue (maintain current activity), Rest (reduce intensity), Increase (raise intensity), Change (switch activity type), and Alert (emergency stop).

Appendix D Flow Matching and FNO Mathematical Details

This section provides the full mathematical formulations for the per-modality Flow Matching (Layer 2) and FNO spectral fusion (Layer 3) that are summarized in the main text (Sections III-C and III-D). We present the conditional flow matching loss, the FNO spectral convolution, the phase-shift representation, and the approximation bound.

Appendix D.1 Conditional Flow Matching Formulation

The flow is defined by the ODE:

$$\frac{d\psi_t(x)}{dt} = v_t(\psi_t(x)), \quad \psi_0(x) = x \tag{A12}$$

Table A19. Parsing Examples

Mission	Final Goal / Interm.	Actions
"traverse the rooms to get to the goal"	goal / [room]	[traverse, get]
"pick up the blue key and open the blue door"	door / [key]	[pick, open]
"find the yellow key then unlock the door to reach the goal"	goal / [key, door]	[find, unlock, reach]

Table A20. Complete Hyperparameter List

Category	Parameter	Value
Architecture	CLIP model	ViT-B/32
	Slot dimension d	128
	Number of slots K	8
	RSSM h_t dimension	256
	RSSM z_t dimension	64
	DNC memory slots N	32
	DNC memory width W	64
Flow Matching	Euler steps N	4
	Velocity MLP hidden dim	128
	Gate init α_0	-3.0
FNO	Fourier modes K	16
	Residual scale init	0.1
Training	Optimizer	Adam
	Learning rate	1×10^{-4}
	Adam ϵ	10^{-5}
	Gradient clip norm	1.0
	PPO clip ϵ	0.2
	PPO epochs per update	4
	GAE λ	0.95
	Discount γ	0.99
Reward	λ_{lang}	0.2
	λ_{VL}	0.1
	$\lambda_{\text{intrinsic}}$	0.2
	λ_{FM} (WM)	0.1
World Model	β_d (KL weight)	0.5
	Free nats	1.0

The CFM training loss for modality m is:

$$\mathcal{L}^m_{\text{CFM}} = \mathbb{E}_{t,z,x_0}\left[\|v^m_\theta(x_t,t) - (z - x_0)\|^2\right] \tag{A13}$$

Appendix D.2 FNO Spectral Convolution

The spectral convolution applies FFT, complex multiplication by learnable weights R_k, and IFFT:

$$\hat{E}_k = \text{FFT}(E)_k \tag{A14}$$

$$E_{\text{fused}} = \text{GELU}(\text{LN}(\text{IFFT}(R_k\hat{E}_k) + WE)) + E \tag{A15}$$

with smoothness regularization:

$$\mathcal{L}_{\text{smooth}} = \lambda_s \sum_{m=1}^{M-1} \|E[m,:] - E[m{+}1,:]\|^2 \tag{A16}$$

Appendix D.3 Phase-Shift Representation

The cross-correlation delay τ between modalities appears as:

$$\mathcal{F}\{\rho_{12}\}(\omega) = S_{12}(\omega) \cdot e^{-j\omega\tau} \tag{A17}$$

Appendix D.4 Approximation Bound

The pre-fusion mutual information bound:

$$I\big((X^1, \ldots, X^M); f(\hat{X}^1, \ldots, \hat{X}^M)\big) \geq I_{\text{post-fusion}} - \mathcal{O}(\sum_m \|\delta^m\|^2/\sigma_m^2) \tag{A18}$$

See Appendix L for the full derivation.

Appendix E Layer 1 Encoder Details

This section provides implementation details for each of the four modality-specific encoders in Layer 1 (Section III-B). We describe the vision encoder (CLIP + Slot Attention with numerical stabilization), the language and state encoders, and the cross-modal attention mechanism that aligns vision and language slots.

Appendix E.1 Vision Encoder

The frozen CLIP ViT-B/32 [11] produces patch features $F_{\text{patch}} \in \mathbb{R}^{49\times 768}$, projected via $F_{\text{vision}} = W_{\text{proj}} F_{\text{patch}} + b_{\text{proj}} \in \mathbb{R}^{49\times d}$. Slot Attention [12] decomposes these into K slots through iterative competitive assignment:

$$A_{ij} = \frac{\exp\big(\frac{1}{\sqrt{d}}(W_q s_i)^\top (W_k F_j)\big)}{\sum_{i'} \exp\big(\frac{1}{\sqrt{d}}(W_q s_{i'})^\top (W_k F_j)\big)} \tag{A19}$$

$$s_i' = \text{GRU}\Big(\sum_j \frac{A_{ij}}{\sum_{j'} A_{ij'}} W_v F_j,\ s_i\Big) \tag{A20}$$

Numerical stabilization techniques: (1) Mask values use -10^9 instead of $-\infty$. (2) NaN fallback substitutes uniform $\mathbf{1}/K$. (3) GRU weights: Xavier (gain 0.5) + Orthogonal (gain 0.5). (4) Variance clipping: $\sigma = \max(\exp(\log \sigma_{\text{raw}}), 10^{-6})$.

Appendix E.2 Language, State, and Reward Encoders

Language: A Transformer encoder [22] produces $H_{\text{lang}} = \text{TransformerEnc}(\text{Embed}(T) + \text{PE})$. Slot Attention [12] is applied to decompose mission components.

State: FNO block [17] with spectral convolution $e_{\text{state}} = \text{IFFT}(R_k \cdot \text{FFT}(o)) + W \cdot o$.

Reward: $e_{\text{reward}} = \text{ReLU}(\text{Conv1D}(r_{t-H:t}))$.

Appendix E.3 Cross-Modal Attention

Bidirectional multi-head attention [22] with residual connections:

$$S'_{\text{vis}} = S_{\text{vis}} + \text{MHA}(S_{\text{vis}}, S_{\text{lang}}, S_{\text{lang}}) \tag{A21}$$

$$S'_{\text{lang}} = S_{\text{lang}} + \text{MHA}(S_{\text{lang}}, S_{\text{vis}}, S_{\text{vis}}) \tag{A22}$$

Appendix F Layer 4 Details

This section provides the implementation details for the DNC memory module and the PPO policy head that comprise Layer 4 of the architecture (Section III-E).

Appendix F.1 DNC Memory

The DNC [18] uses content-based addressing: $w_c(i) = \text{softmax}(\beta \cdot \cos(k, M[i]))$. Write: $M' = M \odot (1 - w_w \mathbf{e}^\top) + w_w \mathbf{a}^\top$. Read: $r = \sum_i w_r(i) M'[i]$. Output: $e_{\text{aug}} = \text{GELU}(W_{\text{out}}[e_{\text{fused}}; r] + b)$.

Appendix F.2 PPO Objective

The PPO [24] clipped surrogate: $\mathcal{L}_{\text{PPO}} = -\mathbb{E}[\min(\rho_t \hat{A}_t, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\hat{A}_t)]$ with $\epsilon = 0.2$ and GAE [40] for advantage estimation.

Appendix G Auxiliary Component Details

This section describes the auxiliary mechanisms that support the main four-layer architecture (Section III-F): language-grounded reward shaping, adaptive reward decay, the success buffer for preventing catastrophic forgetting, and score-based adaptive complexity estimation for FM inference steps.

Appendix G.1 Language-Grounded Reward

Three-component reward [25]: $R_{\text{total}} = \lambda_1 R_{\text{lang}} + \lambda_2 R_{\text{VL}} + \lambda_3 R_{\text{intrinsic}}$. R_{lang}: semantic match between LLM-parsed goals and observations. R_{VL}: cosine similarity [11] between vision and language FM embeddings. $R_{\text{intrinsic}}$: prediction-error curiosity reward [41].

Appendix G.2 Adaptive Reward Shaping

Following the potential-based reward shaping framework [28]: $r_{\text{shaped}} = r_{\text{env}} + \max(1 - 1.25\bar{S}, 0) \cdot r_{\text{bonus}}$, where $\bar{S}$ is the recent success rate.

Appendix G.3 Success Buffer

Inspired by Hindsight Experience Replay [29], successful episodes are stored and replayed at 25% mix ratio during training to prevent catastrophic forgetting.

Appendix G.4 Score-Based Adaptive Complexity

Using the score function from score-based generative models [30]: Complexity $= \mathbb{E}_t[\|s_\theta(x, t, c)\|^2]$ determines the number of FM steps via Gumbel-Softmax [42] selection from $\{1, 2, 3, 5, 7, 10\}$.

Appendix H Crafter: Per-Seed Results

Table A21 reports the per-seed Crafter scores for full transparency. All 10 seeds use 10^6 environment steps with achievement reward shaping enabled. The official Crafter score $(\exp(\frac{1}{22}\sum \ln(1 + s_i)) - 1)$ is computed per seed and then averaged.

Appendix I RSSM Limitations and Input-Side Design

This section analyzes how the standard RSSM architecture introduces structural limitations for multimodal sensor-driven applications, and how FreamerV1's input-side design addresses each limitation without modifying the RSSM's internal dynamics.

Appendix I.1 Addressing RSSM Limitations via Input-Side Design

The standard RSSM architecture [1] processes observations through a single encoder before the deterministic–stochastic state transition, which introduces three structural limitations for multimodal sensor-driven applications. First, a single encoder conflates modalities with fundamentally different noise profiles (e.g., sparse reward signals and high-frequency IMU readings), losing modality-specific structure. Second, the GRU-based deterministic

Table A21. Per-seed Crafter results (official score and mean success rate).

Seed	GPU	Type	Score	Mean SR
0	GPU2[†]	4090	15.9%	19.5%
1	GPU1	4090	16.3%	20.0%
2	GPU2	4090	17.6%	21.3%
3	GPU3	4090	15.3%	18.5%
4	GPU4	4090	16.7%	20.5%
5	GPU5	3090	15.8%	19.5%
6	GPU6	3090	15.8%	19.3%
7	GPU7	3090	17.2%	21.1%
8	GPU8	3090	14.8%	18.1%
9	GPU9	3090	17.0%	21.0%
Mean ± Std			**16.2 ± 0.8%**	**19.9 ± 1.0%**

[†]Seed 0 was trained earlier on the same machine as Seed 2.

pathway provides only short-term memory, insufficient for recalling episodic patterns such as prior heart-rate spikes across activity transitions. Third, standard real-valued fusion (concatenation or attention) cannot represent phase-shifted cross-modal correlations, such as the several-second delay between physical movement (IMU) and heart-rate elevation.

The proposed architecture addresses these limitations not by modifying the RSSM's internal dynamics, but by improving the quality of the observation representation that enters the RSSM posterior. Per-modality Flow Matching preserves modality-specific noise characteristics by denoising each signal independently before fusion. FNO spectral fusion encodes phase-shifted correlations through learnable complex-valued spectral weights (Eq. (A17)). DNC external memory supplements the GRU's short-term state with content-addressable episodic recall. This input-side design is complementary to approaches that modify the RSSM itself, such as MedDreamer's Adaptive Feature Integration module [8] for irregular clinical time series, and could in principle be combined with such internal modifications.

Appendix J PAMAP2 World Model: Full Implementation Details

This section provides the complete implementation details of the PAMAP2 health management application described in Section 5. The system uses a Recurrent State-Space Model (RSSM) as the world model backbone, with the proposed foundation encoder (Slot Attention + per-modality FM + FNO + DNC) as the observation encoder. The agent learns to recommend health interventions (Continue, Adjust Intensity, Rest, Alert) by imagining future physiological trajectories in the latent space, avoiding the ethical and practical difficulties of exposing real subjects to dangerous physiological states during training.

Appendix J.1 RSSM World Model Architecture

The world model follows the RSSM of DreamerV3 [1], representing the latent state s_t as the concatenation of a deterministic component h_t (256 dimensions) and a stochastic component z_t (64 dimensions), yielding a 320-dimensional state vector (Fig. A7).

The deterministic pathway retains temporal context of activity patterns through a GRU:

$$h_{t+1} = \mathrm{GRU}_\theta\big(h_t,\ [z_t;\ e(a_t)]\big) \tag{A23}$$

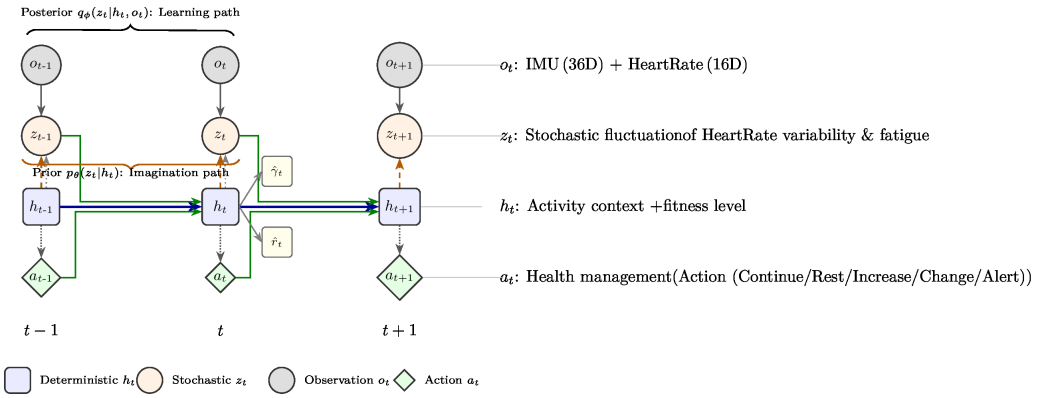

Figure A7. RSSM world model for PAMAP2 health management. Top: temporal unrolling showing the deterministic pathway h_t (GRU transitions) and stochastic pathway z_t (prior/posterior distributions). Bottom: four-phase training pipeline with experimental results.

The stochastic pathway models uncertainty through prior and posterior distributions:

$$\text{Prior:} \quad p_\theta(z_t \mid h_t) = \mathcal{N}\big(\mu_\theta(h_t),\ \sigma_\theta(h_t)\big) \tag{A24}$$

$$\text{Posterior:} \quad q_\phi(z_t \mid h_t, o_t) = \mathcal{N}\big(\mu_\phi(h_t, o_t^{\text{embed}}),\ \sigma_\phi(h_t, o_t^{\text{embed}})\big) \tag{A25}$$

where o_t^{embed} is the output of the foundation encoder (Section 5). The posterior conditions on observations via the encoder, while the prior enables imagination without observations.

Appendix J.2 Foundation Encoder Details

The observation o_t consists of IMU sensors at three body locations (accelerometer, gyroscope, magnetometer; 36 dimensions) and heart-rate-related features (16 dimensions). The foundation encoder applies the four-stage pipeline:

Stage 1 (Slot Attention): The three IMU sites (3×12 dimensions) are treated as slots and processed by Self-Attention, then fused with heart rate features via Cross-Attention:

$$o_t^{\text{slot}} = \text{CrossAttn}\big(\text{SelfAttn}(\text{IMU}_t),\ \text{MLP}(\text{HR}_t)\big) \tag{A26}$$

Stage 2 (Per-Modality Flow Matching): Independent FM modules refine the IMU and HR representations before fusion. Without a pre-trained encoder (no CLIP equivalent for IMU data), the FM refinement compensates for the raw encoder's noise—this is the setting where filter-before-mixing is most critical.

Stage 3 (FNO Spectral Fusion): The refined modality representations are fused via SpectralConv1d. The phase-shift property of the FNO (Eq. (A17)) is particularly relevant here, as physical movement (IMU) precedes heart rate elevation by several seconds.

Stage 4 (DNC Memory): The fused representation is augmented with episodic memory via content-based addressing [18], enabling the agent to recall past physiological episodes (e.g., a prior heart-rate spike that triggered a rest recommendation).

The final encoded observation o_t^{embed} enters the RSSM posterior (Eq. (A25)).

Appendix J.3 World Model Training

The world model is trained by minimizing:

$$\mathcal{L}_{\text{WM}} = \mathcal{L}_{\text{rec}} + \beta_d\, D_{\text{KL}}(q_\phi \| p_\theta) + \mathcal{L}_{\text{rew}} + \mathcal{L}_{\text{con}} \tag{A27}$$

Table A22. Comparison with World Model-Based Health Intervention Systems

	MedDreamer [8]	KANDI [9]	DreamerV3 [1]	**FreamerV1 (ours)**
Domain	EHR (Sepsis)	Wearable (fall risk)	General RL	Wearable (health)
World model	RSSM+AFI	None	RSSM	RSSM+Foundation
Policy	Actor-Critic	Diffusion	Actor-Critic	PPO
Learning	Online+Imag.	Offline IRL	Online+Imag.	Online+Imag.
Encoding	AFI	MLP	CNN/MLP	SlotAttn+FM+FNO
Anomaly	—	—	—	KL divergence

where $\mathcal{L}_{\text{rec}}$ reconstructs observations (MSE on IMU + HR features), $\mathcal{L}_{\text{rew}}$ predicts rewards (health outcomes), $\mathcal{L}_{\text{con}}$ predicts episode continuation (activity transitions), and $\beta_d = 1.0$ balances KL regularization.

Appendix J.4 Imagination-Based Policy Optimization

Using only the prior distribution, imagined rollouts of horizon $H = 15$ are generated to optimize the policy without real-world interaction:

$$\begin{aligned} \hat{h}_{t+1} &= \text{GRU}_\theta(\hat{h}_t,\, [\hat{z}_t;\, e(\hat{a}_t)]) \\ \hat{z}_{t+1} &\sim p_\theta(\cdot \mid \hat{h}_{t+1}), \quad \hat{a}_t \sim \pi_\psi(\cdot \mid \hat{s}_t) \end{aligned} \tag{A28}$$

The policy π_ψ and value function V_ξ are updated via λ-returns computed over imagined trajectories.

The reward function combines physiological targets (heart rate within safe range, activity level maintenance, fatigue prevention) with safety constraints (penalizing prolonged exposure to dangerous heart rate zones >160 bpm). This imagination-based training is particularly valuable for health management: learning appropriate Alert actions requires repeated exposure to high-risk states, which is ethically and practically infeasible in real patients. The world model generates such scenarios safely in the latent space.

Appendix J.5 Physiological Anomaly Detection

As a byproduct of the world model, the KL divergence $D_{\text{KL}}(q_\phi \| p_\theta)$ at each time step quantifies how much the actual sensor observation deviates from the world model's prediction. Time steps where $D_{\text{KL}} > \mu_D + 2\sigma_D$ (where μ_D and σ_D are the running mean and standard deviation of KL values) are flagged as anomaly candidates, indicating unexpected heart rate elevation or atypical fatigue patterns. This mechanism provides an additional safety layer not present in MedDreamer or KANDI (Table A22).

Appendix J.6 Positioning Among Health Intervention Systems

Table A22 compares our approach with recent world model-based health intervention systems. Direct numerical comparison is not meaningful across different clinical domains; the table highlights architectural differences.

Our system shares with MedDreamer the paradigm of RSSM-based imagination for safe policy learning, but addresses a different data regime: continuous, high-frequency, multi-site wearable signals rather than sparse, irregular EHR records. Compared to KANDI, our approach learns an explicit world model and optimizes policies through imagination, whereas KANDI operates offline with pre-collected expert demonstrations.

Table A23. OGM-GE Mode Comparison (MultiRoom-N2-S4, 5000 episodes)

Mode	Seed 0	Seed 1	Seed 2	Mean
Manual α	84%	80%	99%	87.7% ± 8.2%
OGM-GE Mode A	**100%**	**95%**	**89%**	**94.7% ± 4.5%**
OGM-GE Mode B	91%	79%	78%	82.7% ± 5.9%

Appendix J.7 Generalization Across Subjects

After pre-training the world model on data from multiple subjects, the policy can be adapted to new subjects with minimal data. While inter-subject variability exists in resting heart rate and cardiopulmonary capacity, higher-level decision structures—such as recommending rest under high exertion or alerting upon fatigue accumulation—are shared in the latent space of the world model, enabling few-shot adaptation.

Appendix K OGM-GE Gradient Modulation: Mode Analysis

We evaluated two modes of OGM-GE gradient modulation for the per-modality FM residual gate:

Mode A (alpha-only): Modulates only the α^m gradient with an amplified learning rate (scale $\times$ lr_scale, where lr_scale $= 10$). This preserves the stability of the velocity field while accelerating gate adaptation.

Mode B (full-params): Modulates all FM parameters (velocity network + α), matching the original OGM-GE paper [26] which modulates entire encoder gradients.

Mode B (Table A23) underperformed manual α tuning (82.7% vs. 87.7%). Analysis of the learning curves reveals the cause: Mode B destabilizes the FM velocity field by amplifying its gradients based on the FM loss disparity ratio, which interferes with the PPO policy gradients that also flow through the velocity network. Seed 2 exhibited a performance collapse from 76% to 60% between episodes 3800–4000, coinciding with elevated Value Loss (1.86) and stagnant Entropy (1.50–1.60 throughout training, compared to 1.12 for Seed 0). The high Entropy indicates that Mode B prevented the policy from specializing, as the velocity field representations changed too rapidly for the policy to track.

Mode A avoids this instability by restricting gradient modulation to the scalar α^m parameters, which control the FM gate opening without altering the learned velocity field. The lr_scale $= 10$ amplification compensates for the inherently small α gradients observed in the manual-tuning experiments (Section 4.3.5).

Appendix L Derivation of the Approximation Bound

We derive the bound stated in Eq. (A18). Let X^m denote the clean representation of modality m, $\tilde{X}^m = X^m + \epsilon^m$ the noisy encoding with $\epsilon^m \sim \mathcal{N}(0, \sigma_m^2 I)$, and $\hat{X}^m = g^m(\tilde{X}^m)$ the FM-denoised representation with error $\delta^m = \hat{X}^m - \mathbb{E}[X^m|\tilde{X}^m]$.

The mutual information between the clean signals and the pre-fusion representation can be written as:

$$\begin{aligned} &I\big((X^1,\ldots,X^M);\, f(\hat{X}^1,\ldots,\hat{X}^M)\big) \\ &= I\big((X^1,\ldots,X^M);\, f(\mathbb{E}[X^1|\tilde{X}^1]+\delta^1,\ldots,\mathbb{E}[X^M|\tilde{X}^M]+\delta^M)\big) \end{aligned} \tag{A29}$$

When the fusion function f is Lipschitz continuous with constant L_f (satisfied by the FNO with bounded spectral weights), a second-order Taylor expansion around $\delta^m = 0$ yields:

$$I\big((X^1,\ldots,X^M);\, f(\hat{X}^1,\ldots,\hat{X}^M)\big) \\ \geq I\big((X^1,\ldots,X^M);\, f(\mathbb{E}[X^1|\tilde{X}^1],\ldots,\mathbb{E}[X^M|\tilde{X}^M])\big) - L_f^2 \sum_m \mathbb{E}[\|\delta^m\|^2]/(2\sigma_m^2) \tag{A30}$$

The first term equals $I_{\text{post-fusion}}$ evaluated at the Bayes-optimal denoised representations. Since $\mathbb{E}[\|\delta^m\|^2]$ measures how far the per-modality FM is from optimal denoising, the bound shows that pre-fusion denoising is beneficial whenever $\sum_m \|\delta^m\|^2/\sigma_m^2$ is small—i.e., when the FM denoising error is smaller than the original noise, a substantially weaker condition than perfect denoising.

With the residual gate $\sigma(\alpha^m)$, the effective error becomes $\sigma(\alpha^m)\cdot\delta^m$, and the bound tightens to:

$$L_f^2 \sum_m \sigma(\alpha^m)^2 \mathbb{E}[\|\delta^m\|^2]/(2\sigma_m^2) \tag{A31}$$

Since $\sigma(\alpha^m) \to 0$ when the FM velocity field is untrained, the penalty vanishes at initialization, ensuring that the pre-fusion approach is never worse than no denoising during the early stages of training.

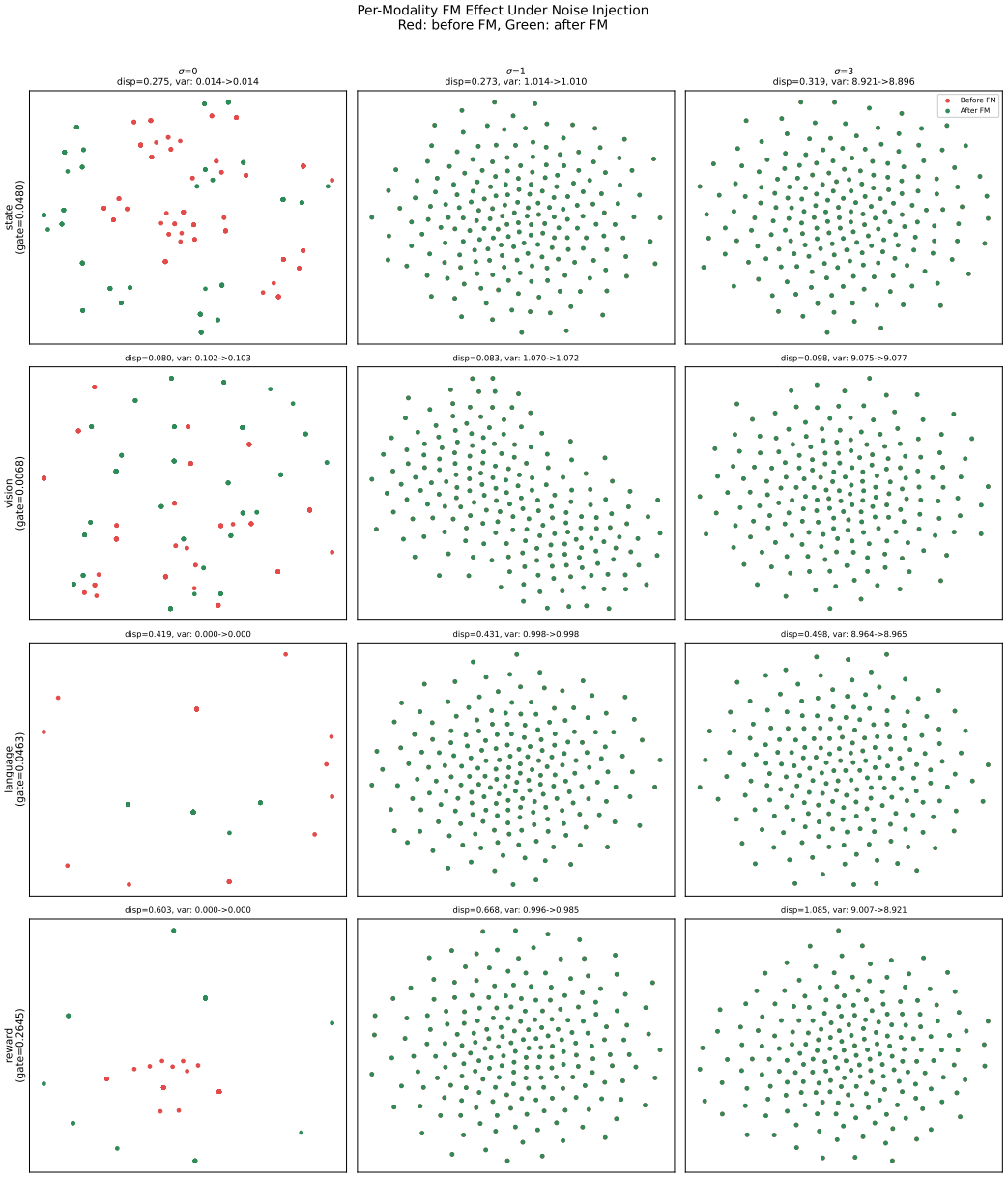

Figure A8. Per-modality FM effect under noise injection ($\sigma \in \{0, 1, 3\}$), visualized via t-SNE. Red: embeddings before FM; green: embeddings after FM. At $\sigma = 0$ (left column), before/after distributions are clearly separated for reward (displacement=0.603) and language (0.419), while vision (0.080) shows minimal change due to its near-closed gate ($\sigma(\alpha) = 0.007$). At $\sigma = 3$ (right column), reward exhibits the largest displacement (1.085) and variance reduction (9.007→8.921), confirming that the gate mechanism concentrates FM refinement on noisy modalities. Vision's gate remains closed even under heavy noise, preserving the high-quality CLIP representations.