COGNITIVE MAP WORLD MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

World models have emerged as a promising paradigm for spatial intelligence, yet current approaches typically treat action-observation pairs as unstructured sequences, neglecting the intrinsic algebraic properties of spatial actions and leading to data inefficiency. We propose the Cognitive Map World Model (CMWM), which integrates the Abelian group structure of spatial actions directly into the latent representation. Inspired by grid cell phase coding, CMWM maps continuous actions to phase shifts across multi-scale periodic modules, constructing a globally cognitive map that eliminates recursive prediction errors and ensures stable spatial memory and imagination. Furthermore, this structured latent space redefines navigation from expensive iterative planning to a congruence solving problem, enabling constant-time trajectory generation. Experiments demonstrate that CMWM significantly outperforms autoregressive and MaskGIT baselines in prediction fidelity, convergence speed, and navigation efficiency.

1 INTRODUCTION

Spatial intelligence—encompassing spatial memory, spatial imagination, and navigation—is fundamental to biological survival and forms a cornerstone of embodied intelligence. Beyond its spatial domain, accumulating evidence suggests that the same neural mechanisms supporting spatial cognition also underlie various higher-order cognitive functions, including temporal cognition Eichenbaum (2014); Tsao et al. (2018), conceptual cognition Constantinescu et al. (2016); Theves et al. (2020), reasoning Stachenfeld et al. (2017); Bellmund et al. (2018), and affective cognition Tavares et al. (2015).

In recent years, World Models have emerged as a powerful paradigm for solving complex spatial intelligence and embodied control tasks Zhou et al. (2024b); Cen et al. (2025); Guo et al. (2025). However, current world models typically treat actions in a simplistic manner, often embedding them without considering their inherent properties Hafner et al. (2025a); Bar et al. (2025). This approach views observation-action data as unstructured sequences, overlooking the implicit structural information introduced by actions. While this affords flexibility in modeling general tasks, it fails to capture the specific regularities of spatial actions.

Our key insight lies in recognizing the intrinsic algebraic structure of spatial actions. For the primary purpose of global positioning and navigation, action sequences generally exhibit four fundamental properties: the existence of an *identity* element (i.e., "no movement"); *invertibility* (e.g., a forward displacement is reversed by a backward one); *associativity* (grouping of actions does not alter the result); and *commutativity* (the order of displacement does not affect the final location) (Fig. 1A). Although general 3D rotations form non-Abelian groups, treating the integration of global pose updates as an Abelian structure allows a world model to represent trajectories more compactly and reduce long-horizon accumulation of prediction errors (Fig. 1 B). Mathematically, actions possessing these properties constitute an *Abelian group*. Any class of actions describable by the same group shares the same underlying structure. Thus, modeling spatial actions inherently involves this Abelian group structure. For continuous, non-periodic environments, this corresponds to the additive group of real numbers $(\mathbb{R}^n, +)$.

However, representing large positive or negative values in the unbounded space $\mathbb{R}^n$ can lead to numerical instability and representation collapse. To address this, we draw inspiration from the mammalian brain. Neuroscience research demonstrates that the brain encodes continuous space using band cells (modeling $\mathbb{R}$) Krupic et al. (2012) and grid cells (modeling $\mathbb{R}^2$) Fyhn et al. (2004).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

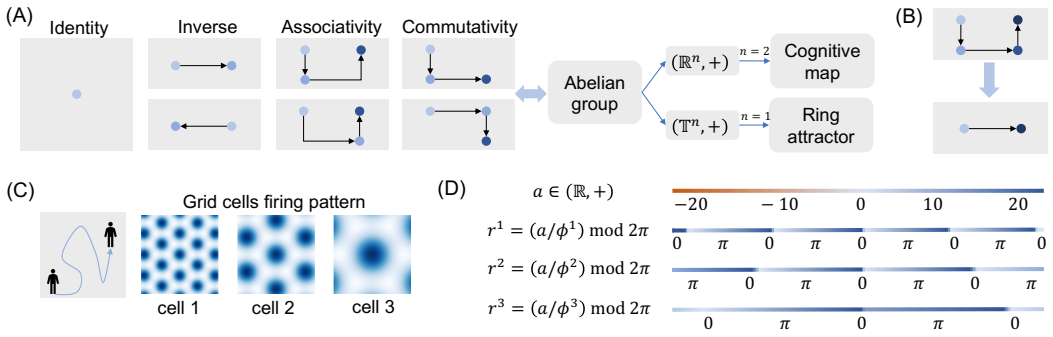

Figure 1: Structured spatial actions and phase coding. (A) The four Abelian-group properties and their neural analogues. (B) Commutativity and invertibility simplify an action sequence from four steps to two. (C) Grid-cell interference patterns at three scales. (D) Multi-period modules provide a bounded representation of continuous space $\mathbb{R}$.

The core principle is *phase coding* in *cognitive map*, where continuous values are mapped onto a set of periodic modules at varying scales. In this framework, spatial actions correspond to phase shifts (rotations). Since each module is periodic, the represented values remain bounded, avoiding magnitude explosion while collectively representing an expansive, continuous space (Fig. 1C).

Based on these insights, we propose the Cognitive Map World Model (CMWM). Architecturally, CMWM follows an encoder-decoder framework (Fig. 2). The observation sequence is encoded into a latent space, where it is aligned with the actual action sequence. During alignment, actions in $\mathbb{R}$ are transformed into phase shifts across multiple periodic modules (Fig. 1D). The aligned latent representation captures both contextual information from observations (combinations across modules) and structural information from actions (phase differences within the same module over time), resembling the band cell-based representation of cognitive maps in the brain. The aligned latent representation is then passed to a decoder for reconstruction, enabling the model to learn this structured representation.

We experimentally demonstrate that CMWM outperforms previous autoregressive (AR) He et al. (2025) and MaskGIT Bruce et al. (2024) models in tasks such as spatial memory and spatial imagination, particularly in long-range prediction, where it shows significant advantages. Additionally, CMWM exhibits faster training stability and convergence, requires fewer inference steps, and possesses scaling capabilities. Moreover, the structured latent map learned by CMWM facilitates highly efficient performance across a wide range of downstream tasks, such as navigation.

2 Related Work

World Models Schmidhuber (1990); LeCun (2022) provide a framework for predicting future observations conditioned on actions. Existing world models can be broadly categorized into two classes based on their operational principles. The first class is autoregressive (AR) models, which predict the next observation step-by-step based on past observations and actions. These models can be implemented using various architectures, including RNNs Chung et al. (2015); Hafner et al. (2019; 2025a), Transformers Chen et al. (2022); Micheli et al. (2022); He et al. (2025), or S4 models Samsami et al. (2024). More recently, conditional diffusion models have also been employed to model the conditional probability distribution Feng et al. (2024); Che et al. (2024); Bar et al. (2025). This formulation naturally enables autoregressive generation by sequentially conditioning on previously generated content. The second category consists of MaskGIT-style models Chang et al. (2022), which model the relationship between actions and observations by predicting randomly masked parts of the observation sequence. Both Transformer-based Bruce et al. (2024) and diffusion-based Hafner et al. (2025b) implementations have been proposed. Unlike autoregressive models, these approaches do not enforce a strict temporal ordering; instead, they generate all masked observations in parallel. Notably, if a Transformer is used as the building block, these paradigms correspond to distinct architectures: AR models typically adopt a GPT-style decoder-only structure, while MaskGIT inherits a BERT-style encoder-only structure. In contrast, our proposed CMWM introduces a novel paradigm that differs from both and is architecturally realized as an encoder-decoder model.

Action Embedding. Both autoregressive and MaskGIT-style models typically embed actions in a relatively simplistic manner, primarily through two common techniques. One approach treats actions as discrete tokens Micheli et al. (2022); Cen et al. (2025), concatenating them with observation tokens to form a single sequence, analogous to language modeling. Alternatively, actions can be integrated as conditioning signals via methods like adaptive layer normalization (AdaLN) Bruce et al. (2024); Bar et al. (2025); He et al. (2025), as introduced by Xu et al. (2019). In contrast, CMWM aligns the encoder-derived latent representations with the action sequence, thereby directly embedding action information into the aligned latent space and yielding latent representations that explicitly encode action structure.

Cognitive Map Models. Recent advances Dong et al. (2025) used the concept of place cells to structure the latent state in an energy-based model. Further work Chandra et al. (2025) proposes that structured latent spaces can be mapped biologically onto the entorhinal–hippocampal loop. However, these implementations often rely on biologically constrained online learning rules, which limits their scalability and practical performance. More recent studies Peng et al. (2025), models the latent space as a structured vector quantized (VQ) representation, enabling parallel data processing. Yet, the discrete nature of the VQ operation restricts its application to discrete actions and hinders generalization to continuous ones. Our CMWM supports parallel modeling of continuous actions, offering greater flexibility and scalability in practical applications.

3 CMWM

Problem Description and Formulation. Our goal is to construct a world model from the agent's exploration data $\{o_0, a_1, o_1, a_2, o_2, \dots\}$. This world model is expected to support spatial memory and spatial imagination tasks, while the learned representations should serve as a basis for downstream spatial tasks such as navigation or model predictive control. The observation is typically a high-dimensional sensory input, such as RGB images, depth maps, or point clouds. The action denotes the agent's movement signal, such as the agent's translation in physical space or control commands of a robotic arm. Throughout this paper, subscripts denote time indices, while superscripts denote component indices within the same time step.

Action Transformation. As discussed in the introduction, we first transform actions into phase representations across periodic modules of different scales. For actions defined in a high-dimensional space $a \in \mathbb{R}^{n_1} \times \mathbb{T}^{n_2}$, we decompose the non-periodic components $\mathbb{R}^{n_1}$ into n_1 individual dimensions $(a^1, \dots, a^{n_1})$. For each $a^j \in \mathbb{R}$, we map it to phase representations across m periodic modules with different periods (Fig. 1D):

$$a^j \to (r^{j,1}, \dots, r^{j,m}), \quad \text{where} \quad r^{j,i} = \frac{a^j}{\phi^i} \mod 2\pi, \tag{1}$$

where $j = 1, \dots, n_1$. Here, a larger ϕ^i corresponds to a module with a longer period. Each a^j is uniquely represented by the set $(r^{j,i})_{i=1}^m$. The composite period of this set of modules is the least common multiple (LCM) of all ϕ^i. In practice, this LCM far exceeds the range of the actions we model, so global periodicity is not a concern. Notably, if any two periods ϕ^i and ϕ^j are incommensurate, the overall system lacks a finite LCM, ensuring that the mapping from the module phases back to the original action is injective. For the inherently periodic components $(a^{n_1+1}, \dots, a^{n_1+n_2})$ in $\mathbb{T}^{n_2}$, each dimension is already periodic and can be directly used as a phase representation. To improve neural network training, we also expand each $a^j \in \mathbb{T}$ into an m-dimensional phase vector:

$$a^j \to (r^{j,1}, \dots, r^{j,m}), \quad \text{where} \quad r^{j,i} = \frac{a^j}{\phi} \cdot 2\pi, \tag{2}$$

where $j = n_1 + 1, \dots, n_1 + n_2$ and ϕ is the intrinsic period of the motion.

Model Architecture and Training. The core idea of CMWM is to embed observation sequences into a latent space that inherently encodes the structure of transformed actions, allowing downstream tasks to benefit from these structured representations within a pretrained backbone. The CMWM is built upon an encoder-decoder framework (Fig. 2). The encoder processes a sequence of observations $o_{0:T}$ and transforms them into latent representations $s_{0:T}$, where each $s_t \in \mathbb{R}^{2n}$.

To ensure that transformations in the latent space directly reflect the structure of actions, we convert the latent representations into phase representations compatible with the action space. Specifically,

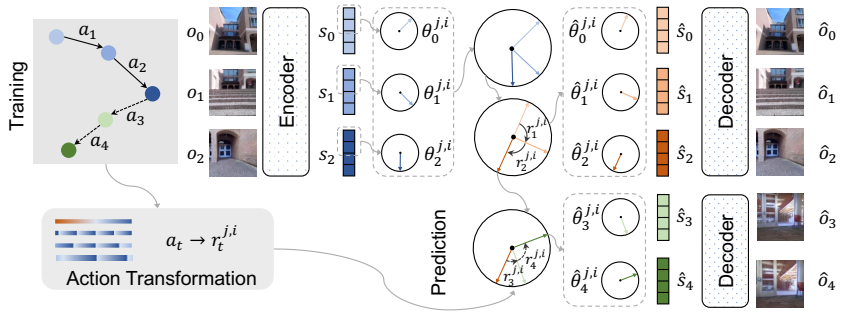

Figure 2: Schematic of the CMWM. The top section depicts the training procedure with $T = 2$; the bottom section illustrates the prediction process for $c = 2, p = 2$.

we interpret each pair of elements in s_t as representing a phase angle, yielding n phases at each time step:

$$\theta_t^k = \text{atan2}(s_t^{2k+1}, s_t^{2k}). \tag{3}$$

The n phases form two functional groups. The first $n_1 m + n_2 m$ action-relevant phases encode motion components ($n_1 m$ aperiodic and $n_2 m$ periodic), while the remaining action-invariant phases encode context independent of movement.

We reindex the action-relevant phases as $\theta_t^{j,i}$, where $j = 1, \ldots, n_1 + n_2$ denotes the motion dimension and $i = 1, \ldots, m$ denotes the module index (blue arrows in Fig. 2). The fundamental constraint we impose is that phase transitions should be governed by actions:

$$\hat{\theta}_t^{j,i} = \hat{\theta}_{t-1}^{j,i} + r_t^{j,i} \tag{4}$$

However, the raw phases $\theta_t^{j,i}$ derived from the encoder do not inherently satisfy this constraint. We therefore seek aligned phases $\hat{\theta}_t^{j,i}$ that satisfy the action constraint while minimizing deviation from the original phases. The aligned phase sequence (orange arrows in Fig. 2) is obtained by solving the following optimization problem:

$$\hat{\theta}_{0:T}^{j,i} = \arg\min_{\hat{\theta}_{0:T}^{j,i}} \sum_{t=0}^{T} d\left(\hat{\theta}_t^{j,i},\ \theta_t^{j,i}\right) \tag{5}$$

where the circular distance metric is defined as:

$$d(\theta_1, \theta_2) = (\cos\theta_1 - \cos\theta_2)^2 + (\sin\theta_1 - \sin\theta_2)^2 \tag{6}$$

Let $\rho_k^{j,i} = \sum_{\tau=1}^{k} r_\tau^{j,i}$. This optimization problem admits the following globally optimal closed-form solution:

$$\hat{\theta}_t^{j,i} = \text{atan2}\left(\sum_{k=0}^{T} \sin(\theta_k^{j,i} - \rho_k^{j,i}), \sum_{k=0}^{T} \cos(\theta_k^{j,i} - \rho_k^{j,i})\right) + \rho_t^{j,i}. \tag{7}$$

The rigorous derivation is provided in Appendix A.

For the action-invariant phases, we treat the action signals as zero ($r_t = 0$). Substituting this into Eq.(7) yields a temporal average of the original phases:

$$\hat{\theta}_t^{j,i} = \text{atan2}\left(\sum_{k=0}^{T} \sin\theta_k^{j,i}, \sum_{k=0}^{T} \cos\theta_k^{j,i}\right). \tag{8}$$

The aligned latent representation $\hat{\theta}_t$, which combines both the action-relevant phases and the action-invariant phases, is then mapped back to the latent space according to their pervious index:

$$\hat{s}_t^{2k} = \cos\hat{\theta}_t^k, \quad \hat{s}_t^{2k+1} = \sin\hat{\theta}_t^k. \tag{9}$$

The resulting latent representation $\hat{s}_t$ is passed to the decoder to reconstruct the original observations $\hat{o}_t$. Since the mapping from θ_t to $\hat{\theta}_t$ involves discontinuous operations, we employ the straight-through

estimator Van Den Oord et al. (2017) during training, passing gradients directly from the aligned to the original phases. Additionally, we introduce a commitment loss to regularize the alignment. The complete training objective is:

$$\mathcal{L} = \|o_{0:T} - \hat{o}_{0:T}\|^2 + \beta \left\| \theta^{1:n}_{0:T} - \text{sg}\left[\hat{\theta}^{1:n}_{0:T}\right] \right\|^2 \tag{10}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operation and β adjusts the strength of the commitment loss. This approach ensures that the learned representation captures both the spatial structure and the contextual information necessary for accurate prediction.

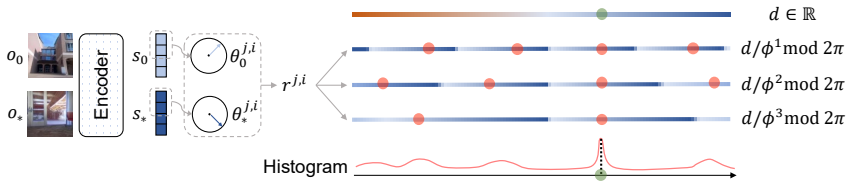

Figure 3: Navigation via congruence solving in the latent phase space. The model encodes start (o_0) and target (o_*) observations to compute phase differences. See Figure 8 in Appendix E.3 for a concrete experimental demonstration.

Spatial Memory and Imagination. After training the CMWM, it can be employed for prediction tasks. Predictions made on the training set are termed spatial memory, while those on the test set are termed spatial imagination. Specifically, given an initial sequence of observations and actions $\{o_{0:c}, a_{1:c}\}$ and a target action sequence $\{a_{c+1:c+p}\}$ (indicated by dashed lines in Fig. 2), the model predicts the resulting observations $\{o_{c+1:c+p}\}$.

The prediction procedure consists of two key steps. Step 1: This step mirrors the training process. The observation sequence is encoded and aligned in the latent space (orange circles in Fig. 2). When only a single initial observation is provided, alignment is unnecessary. This process yields the aligned phase representation $\hat{\theta}_c$. Step 2: The target action sequence $\{a_{c+1:c+p}\}$ is transformed into phase increments via the action transformation module. These increments are then sequentially applied to the initialized phases:

$$\hat{\theta}^{j,i}_t = \hat{\theta}^{j,i}_{t-1} + r^{j,i}_t \quad \text{for} \quad t = c+1, \ldots, c+p \tag{11}$$

resulting in the propagated phases (green arrows in Fig. 2). The final predicted observations $\{\hat{o}_{c+1:c+p+1}\}$ are generated by decoding the propagated phases through the decoder. This framework enables the model to simulate future states by leveraging a structured latent space that inherently captures the properties of spatial transformations.

Navigation as Congruence Solving. In prior world model approaches, navigation is typically formulated as a model predictive control (MPC) problem. However, since CMWM possesses a structured latent representation that directly relates to actions, our navigation framework transforms into solving a congruence problem in phase space, significantly reducing computational complexity.

Given a starting observation o_0 and a navigation target o_*, the objective is to find an action sequence $(a_1, \ldots, a_T)$ to reach o_*. The world model predicts the resulting observation o_T after executing actions $(a_1, \ldots, a_T)$ from o_0, making o_T a function of both the actions and initial state. Navigation thus reduces to the optimization problem:

$$\arg\min_{a_{1:T}} \|o_T(a_{1:T}, o_0) - o_*\|^2 \tag{12}$$

This constitutes a classical MPC problem. Solving it via gradient descent is computationally intensive and prone to convergence issues. Consequently, previous works Hansen et al. (2023); Zhou et al. (2024a); Bar et al. (2025) employ the Cross Entropy Method (CEM) Rubinstein (1997), a derivative-free, population-based optimization technique. Nevertheless, these approaches still require multiple iterations and extensive world model evaluations, incurring substantial computational overhead.

Human navigation does not involve pre-optimizing every detailed action in advance. Instead, we naturally first determine the direction of the target location, then plan more detailed movements toward that direction. CMWM's navigation strategy mirrors this approach by first determining the target direction of o_* through the latent state.

Specifically, given starting observation o_0 and target observation o_*, for the Euclidean components of the action (e.g., position in 2D space), assuming the difference along dimension j is d^j, the phase difference $r^{j,i}$ according to the training constraint in Eq. (4) should be:

$$r^{j,i} \approx \theta_*^{j,i} - \theta_0^{j,i} \tag{13}$$

where $\theta_*^{j,i}$ and $\theta_0^{j,i}$ are the phase representations obtained by encoding o_0 and o_* respectively. This $r^{j,i}$ can be computed from the above equation. Furthermore, from the action transformation formula in Eq. (1), we know the relationship between the actual difference d^j and the phase difference $r^{j,i}$ is:

$$r^{j,i} = \frac{d^j}{\phi^i} \mod 2\pi \tag{14}$$

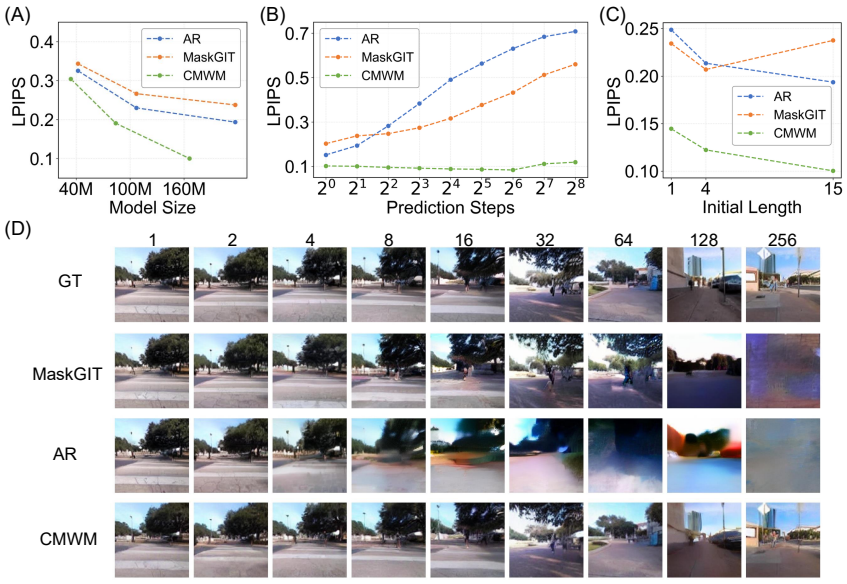

Figure 4: Scaling and long-horizon prediction. LPIPS (lower is better) versus (A) model size, (B) prediction horizon, and (C) context length. (D) Ground truth and predictions from MaskGIT, AR, and CMWM; CMWM remains stable as the horizon grows.

Our task is to determine d^j from the m equations for $r^{j,i}$. While d^j uniquely determines $r^{j,i}$, the converse is not true—multiple values of d^j can produce the same $r^{j,i}$ (analogous to multiple red points along a line in Fig. 3). However, when we have m phase measurements and at least two periods are incommensurate, d^j is uniquely determined. To solve for d^j, we generate candidate solutions $d^{j,i,k} = (2\pi k + r^{j,i})\phi^i,\ k \in \mathbb{Z}$ for each module i within a plausible range, then employ histogram statistics to identify the value with the highest frequency, which corresponds to the true d^j (Fig 3). Interestingly, the brain employs a similar mechanism for direction estimation from grid cell activities Bush et al. (2015). The same methodology can be applied to periodic components $\mathbb{T}$ (e.g., head direction) to determine their differences d^j.

4 Experiment

As a world model, we evaluated CMWM on spatial memory, imagination, and navigation tasks. The results demonstrate that CMWM holds significant advantages in long-range prediction and exhibits strong performance in long-distance navigation.

Datasets. We utilized the robotics datasets SCAND Karnan et al. (2022), TartanDrive Triest et al. (2022), HuRoN Hirose et al. (2023) and RECON Shah et al. (2021) for training and evaluation. Following prior work Sridhar et al. (2024), we preprocessed the datasets, which consist of observation-action sequences collected during a robot's free exploration in natural environments. The observation $o \in \mathbb{R}^{128\times128\times3}$ corresponds to RGB input from a camera. In our experiments, adhering to standard practice in world modeling, we first employ a pretrained VQ-VAE encoder Rombach et al. (2022)

to compress o into a latent representation of size $\mathbb{R}^{16\times16\times4}$. The CMWM operates on these compressed observations. For visualization, the latent representations are decoded back into the original pixel space using the VQ-VAE decoder. The action is represented as $a = (a^1, a^2, a^3)$, where the components denote displacements along the x and y axes $(a^1, a^2) \in \mathbb{R}^2$, and the head direction $a^3 \in \mathbb{T}$.

Evaluation Metrics. For spatial memory and spatial imagination tasks, we employ LPIPS Zhang et al. (2018) to measure perceptual similarity through deep feature comparisons and PSNR for pixel-level quality assessment at generated positions with ground truth (GT). To evaluate synthesis quality, we use FID Heusel et al. (2017) to assess the distribution of generated data. For navigation tasks, we measure the distance between the predicted position and the ground truth position as the evaluation criterion.

Implementation Details. Both the encoder and decoder in our model are implemented using ViTs Dosovitskiy et al. (2020) (see Appendix B). In our ablation studies, we will examine the impact on model performance when employing only spatial attention versus combining both spatial and temporal attention mechanisms (see Appendix D).

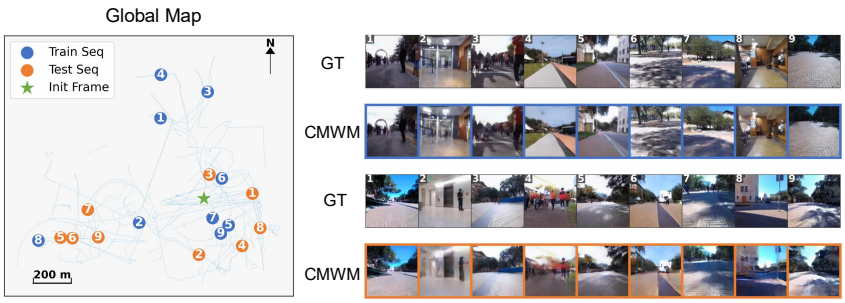

Figure 5: Global-scale spatial memory and imagination. Left: trajectories, training locations (blue), test locations (orange), and initialization (green star). Right: ground truth and CMWM predictions.

Baselines. We employed both AR-style He et al. (2025) and MaskGIT-style Bruce et al. (2024) world models as our baseline methods. To ensure a fair comparison, all baselines were constructed using the same ViT architecture as CMWM, incorporating both spatial and temporal attention mechanisms as their building blocks. For action embedding, both AR and MaskGIT baselines employ AdaLN to condition actions into each layer of the ViT, as per their original implementations. The baseline models also utilized the same pre-trained VQ-VAE to process RGB images, with model inputs identical to those of CMWM. For navigation tasks, we followed the established practice Bar et al. (2025) of using the CEM for planning.

Spatial Memory and Imagination. We evaluated CMWM against strong baselines (AR and MaskGIT) on two core capabilities. Both tasks follow the same workflow: predicting a sequence of future observations given an initial observation sequence and an arbitrary action sequence. Spatial Memory evaluates predictions for visited locations using initializations from the training set, while Spatial Imagination evaluates predictions for unvisited locations using initializations from the test set. Performance was measured using PSNR, LPIPS, and FID.

Visual comparisons of long-horizon generation on the SCAND dataset are presented in Figure 4. As illustrated in Figure 4D, the quality of images generated by the AR baseline degrades rapidly as the prediction steps increase. By step 32, the AR model exhibits severe blurring and loss of geometric detail, a classic symptom of error accumulation inherent in recursive prediction schemes. MaskGIT shows improved stability but still suffers from visual artifacts and inconsistency at longer horizons.

In contrast, CMWM maintains high-fidelity generation throughout the entire sequence, preserving sharp structural details even at step 256. This stability stems from our model's unique architecture: instead of predicting o_{t+1} based on o_t, CMWM calculates the latent state at any time step t directly by applying the accumulated action phase shift to the initial state. This demonstrates that the model has constructed a stable representation of the real world.

Figure 5 further visualizes the model's spatial capability on a global scale. The left panel displays the trajectories for both training and testing sequences within the global map. The qualitative samples on the right demonstrate that CMWM can accurately recall complex visual details from visited locations

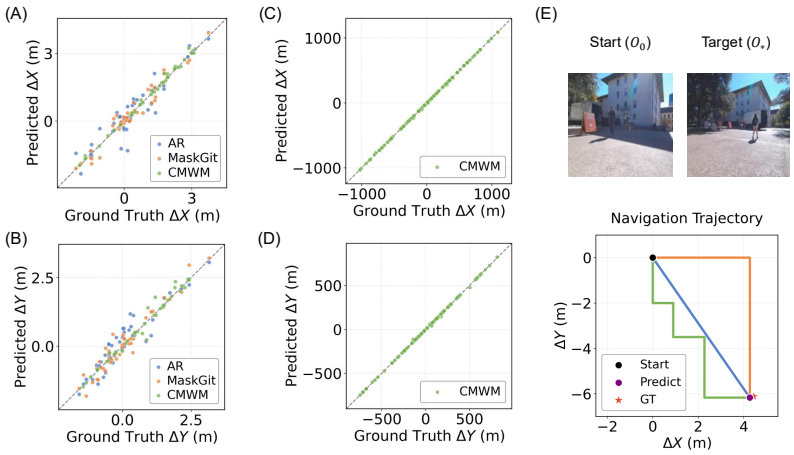

Figure 6: Navigation results. (A-B) Predicted versus ground-truth short-range displacement for AR, MaskGIT, and CMWM. (C-D) CMWM remains accurate over distances up to 1000 m. (E) Direct, axis-aligned, and zigzag trajectories generated between o_0 and o_*.

(Memory) and generalize this structural understanding to generate realistic views for unvisited places (Imagination), effectively functioning as a neural cognitive map. Such ultra-long-distance generation is unattainable for AR and MaskGIT baselines, which require step-by-step generation to traverse the path, imposing severe challenges on both prediction accuracy and inference time. In contrast, CMWM mimics human mental imagination by directly transitioning to the target state in a single step via our designed latent phase rotation, independent of the travel distance. Furthermore, we provide empirical evidence of CMWM's capability on multiple maps (see Appendix E.2).

We report quantitative metrics (PSNR, LPIPS, and FID) for the Training set (Memory) in Table 5 and the Test set (Imagination) in Table 6 (see Appendix E.1). In the Spatial Memory task, CMWM consistently outperforms both baselines across all time steps. Notably, the performance gap widens significantly as the sequence length increases. In the Spatial Imagination task, CMWM demonstrates superior generalization capabilities, surpassing baselines in most metrics. The only exception lies in short-range FID scores, where MaskGIT shows competitive performance. However, this is because FID primarily measures the visual plausibility of generated images rather than their alignment with the specific ground-truth locations. MaskGIT tends to generate visually realistic images that are, however, completely misplaced relative to the actual target positions.

Figure 4A and B illustrate the scaling laws of the models regarding model size and prediction steps. We observe that scaling the model size improves performance across all architectures. As prediction steps increase, the LPIPS score for the AR and MaskGIT model deteriorates as expected. In contrast, CMWM exhibits remarkable stability, maintaining consistent generation quality regardless of the prediction horizon. Furthermore, in Figure 4C, we investigate the impact of varying the initial sequence length, where CMWM consistently achieves the best performance.

The performance gap highlights a fundamental difference in mechanisms: AR and MaskGIT must memorize long trajectories to maintain consistency, leading to recursive error accumulation. In contrast, CMWM succeeds by constructing a stable cognitive map. This confirms that for spatial tasks, structural inductive bias is far more data-efficient than relying solely on unstructured sequence modeling.

Navigation. Unlike traditional world models that require iterative rollouts to search for a path, CMWM formulates navigation as solving a congruence problem in the phase space. We tested the model's ability to estimate the relative displacement $(\Delta X, \Delta Y)$ between pairs of observations, from a starting observation o_0 to a target observation o_*. Figure 6 presents the correlation between the predicted displacement and the ground truth. Figure 6A&B show that AR and MaskGIT struggle to accurately estimate spatial relationships, particularly for larger distances. In contrast, CMWM exhibits a near-perfect linear correlation for both ΔX and ΔY axes, maintaining remarkable precision even over extended distances (Figure 6C&D). This indicates that our model has successfully encoded the global coordinates into its latent phase representation, enabling it to precisely calculate the vector difference between any two observed locations purely from their latent states. Leveraging

this capability, the model can generate specific action trajectories tailored to distinct requirements to complete the navigation task, as illustrated by the diverse paths in Figure 6E. For quantitative results, please refer to Table 7 (Appendix E.1) .

The most prominent advantage of CMWM lies in its computational efficiency, as detailed in Table 1. Traditional world models (AR and MaskGIT) typically treat navigation as a model-based planning problem, employing iterative optimization algorithms like CEM to search for optimal actions. This approach incurs a heavy computational burden, as the total inference cost scales linearly with both the planning horizon T and the number of optimization iterations N_{CEM}. Our model fundamentally redefines the navigation paradigm. Since the relationship between states and actions is explicitly encoded via the Abelian group structure, navigation is reduced to a single-step calculation. Consequently, the number of required forward passes is effectively constant, independent of the distance to the target. This represents an orders-of-magnitude acceleration compared to planning-based methods, making CMWM uniquely capable of supporting high-frequency, real-time navigation. Also in complex environments, CMWM can serve as a global planner to guide a local planner, enabling hierarchical navigation that combines long-horizon efficiency with local precision.

	MaskGIT	AR	CMWM
Prediction	$N_{\text{MaskGIT}}T$	T	1
Navigation	$N_{\text{CEM}}N_{\text{MaskGIT}}T$	$N_{\text{CEM}}T$	1

Table 1: Forward-pass complexity. N_{MaskGIT} is the number of MaskGIT refinement passes; N_{CEM} is the number of CEM iterations times candidates (typically about 120).

5 Discussion

In this work, we introduced CMWM, a novel framework that integrates the intrinsic algebraic structure of spatial actions into latent representations. By employing multi-scale phase coding inspired by the brain's grid cell system, CMWM constructs a globally consistent cognitive map. Such a global map can serve as a spatial backbone for future compositional world models. This approach not only ensures stability in long-horizon spatial memory and imagination but also fundamentally simplifies navigation by reducing it to a congruence solving problem.

A critical distinction between CMWM and prior world models lies in our dual focus on latent space utility and structural constraints. First, while recent trends often treat world modeling as a pure sequence generation task, we argue that high-fidelity generation alone is insufficient for embodied intelligence; instead, we prioritize learning a high-quality latent space that explicitly supports downstream control tasks. Second, traditional models attempt to implicitly learn the underlying group structure of the world from massive data. However, without constraints, models with strong generalization capabilities are prone to generating results that violate this structure. By explicitly modeling this group structure, CMWM ensures that the generated world states adhere to physical laws, avoiding the inconsistencies common in unconstrained data-driven approaches.

Looking ahead, our current implementation relies solely on RGB inputs. A natural extension is to incorporate depth information into the observation space. Integrating depth would enable the model to construct a geometrically grounded map that explicitly encodes 3D structures. Such a representation would extend the system's capabilities from pure localization to comprehensive spatial understanding, allowing for obstacle avoidance and more sophisticated trajectory planning in complex environments.

AI Use Statement

Generative AI tools assisted in organizing the authors' model description, drafting manuscript text, preparing the LaTeX project. The authors reviewed all AI-assisted text and figures for accuracy and consistency with the implemented method. The authors take responsibility for the final content of this work, including all AI-assisted text and figures.

Reproducibility Statement

The model formulation, phase-alignment procedure, architecture, training configuration, ablations, and additional quantitative results are provided in the main paper and appendices. The appendices report optimization settings, compute resources, and hyperparameters needed to reproduce the experiments. The submission is anonymized for double-blind review.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15791–15801, 2025.

Jacob LS Bellmund, Peter Gärdenfors, Edvard I Moser, and Christian F Doeller. Navigating cognition: Spatial codes for human thinking. *Science*, 362(6415):eaat6766, 2018.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Daniel Bush, Caswell Barry, Daniel Manson, and Neil Burgess. Using grid cells for navigation. *Neuron*, 87(3):507–520, 2015.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Sarthak Chandra, Sugandha Sharma, Rishidev Chaudhuri, and Ila Fiete. Episodic and associative memory from spatial scaffolds in the hippocampus. *Nature*, 638(8051):739–751, 2025.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11315–11325, 2022.

Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. Gamegen-x: Interactive open-world game video generation. *arXiv preprint arXiv:2411.00769*, 2024.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015.

Alexandra O Constantinescu, Jill X O'Reilly, and Timothy EJ Behrens. Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468, 2016.

Xingsi Dong, Xiangyuan Peng, and Si Wu. Predictive learning in energy-based models with attractor structures. *arXiv preprint arXiv:2501.13997*, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Howard Eichenbaum. Time cells in the hippocampus: a new dimension for mapping memories. *Nature Reviews Neuroscience*, 15(11):732–744, 2014.

Ruili Feng, Han Zhang, Zhantao Yang, Jie Xiao, Zhilei Shu, Zhiheng Liu, Andy Zheng, Yukun Huang, Yu Liu, and Hongyang Zhang. The matrix: Infinite-horizon world generation with real-time moving control. *arXiv preprint arXiv:2412.03568*, 2024.

Marianne Fyhn, Sturla Molden, Menno P Witter, Edvard I Moser, and May-Britt Moser. Spatial representation in the entorhinal cortex. *Science*, 305(5688):1258–1264, 2004.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pp. 1–7, 2025a.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025b.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

Haoran He, Yang Zhang, Liang Lin, Zhongwen Xu, and Ling Pan. Pre-trained video generative models as world simulators. *arXiv preprint arXiv:2502.07825*, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Sacson: Scalable autonomous control for social navigation. *IEEE Robotics and Automation Letters*, 9(1):49–56, 2023.

Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7(4): 11807–11814, 2022.

Julija Krupic, Neil Burgess, and John O'Keefe. Neural representations of location composed of spatially periodic bands. *Science*, 337(6096):853–857, 2012.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1), 2022.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Xiangyuan Peng, Xingsi Dong, and Si Wu. Vector quantization in the brain: Grid-like codes in world models. *arXiv preprint arXiv:2510.16039*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Reuven Y Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.

Mohammad Reza Samsami, Artem Zholus, Janarthanan Rajendran, and Sarath Chandar. Mastering memory tasks with world models. *arXiv preprint arXiv:2403.04253*, 2024.

Jürgen Schmidhuber. *Making the world differentiable: on using self supervised fully recurrent neural networks for dynamic reinforcement learning and planning in non-stationary environments*, volume 126. Inst. für Informatik, 1990.

Dhruv Shah, Benjamin Eysenbach, Nicholas Rhinehart, and Sergey Levine. Rapid Exploration for Open-World Navigation with Latent Goal Models. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=d_SWJhyKfVw.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70. IEEE, 2024.

Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643–1653, 2017.

Rita Morais Tavares, Avi Mendelsohn, Yael Grossman, Christian Hamilton Williams, Matthew Shapiro, Yaacov Trope, and Daniela Schiller. A map for social navigation in the human brain. *Neuron*, 87(1):231–243, 2015.

Stephanie Theves, Guillén Fernández, and Christian F Doeller. The hippocampus maps concept space, not feature space. *Journal of Neuroscience*, 40(38):7318–7325, 2020.

Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. Tartandrive: A large-scale dataset for learning off-road dynamics models. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2546–2552. IEEE, 2022.

Albert Tsao, Jørgen Sugar, Li Lu, Cheng Wang, James J Knierim, May-Britt Moser, and Edvard I Moser. Integrating time from experience in the lateral entorhinal cortex. *Nature*, 561(7721):57–62, 2018.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Xue-Xin Wei, Jason Prentice, and Vijay Balasubramanian. A principle of economy predicts the functional architecture of grid cells. *eLife*, 4, September 2015. ISSN 2050-084X. doi: 10.7554/elife.08362. URL http://dx.doi.org/10.7554/eLife.08362.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. *Advances in neural information processing systems*, 32, 2019.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024a.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024b.

A Derivation of Optimal Phase Alignment (Eq. 7)

In this section, we provide the detailed derivation for the closed-form solution of the phase alignment problem presented in Eq. 7 of the main paper.

A.1 Problem Formulation

Recall the optimization objective defined in Eq. 5. We aim to find the aligned phase sequence $\hat{\theta}^{j,i}_{0:T}$ that minimizes the cumulative distance to the encoder-derived phases $\theta^{j,i}_{0:T}$, subject to the action constraints. For brevity, we omit the superscripts j, i in the following derivation, as the optimization is performed independently for each phase dimension.

The objective function is:

$$\mathcal{L} = \sum_{t=0}^{T} d(\hat{\theta}_t, \theta_t), \tag{15}$$

where the distance metric $d(\cdot, \cdot)$ is the squared Euclidean distance in the embedding space (Eq. 6):

$$d(\hat{\theta}_t, \theta_t) = (\cos\hat{\theta}_t - \cos\theta_t)^2 + (\sin\hat{\theta}_t - \sin\theta_t)^2. \tag{16}$$

Expanding the squared terms, the distance simplifies to:

$$d(\hat{\theta}_t, \theta_t) = 2 - 2\cos(\hat{\theta}_t - \theta_t). \tag{17}$$

Thus, minimizing the cumulative distance $\mathcal{L}$ is equivalent to maximizing the cumulative cosine similarity:

$$\max_{\hat{\theta}_{0:T}} \sum_{t=0}^{T} \cos(\hat{\theta}_t - \theta_t). \tag{18}$$

A.2 Constraint and Transformation

The aligned phases must strictly satisfy the action transition constraint (Eq. 4). For any time step t, the phase $\hat{\theta}_t$ is determined by the initial phase $\hat{\theta}_0$ and the cumulative action phase shifts:

$$\hat{\theta}_t = \hat{\theta}_0 + \mathcal{R}_t, \quad \text{where} \quad \mathcal{R}_t = \sum_{\tau=1}^{t} r_\tau, \tag{19}$$

and $\mathcal{R}_0 = 0$. Substituting this into Eq. 18, the optimization problem reduces to finding a single scalar $\hat{\theta}_0$:

$$\hat{\theta}_0^* = \arg\max_{\hat{\theta}_0} \sum_{t=0}^{T} \cos(\hat{\theta}_0 + \mathcal{R}_t - \theta_t). \tag{20}$$

Let $\phi_t = \theta_t - \mathcal{R}_t$. We can interpret ϕ_t as the *back-projected phase*, representing what the phase at time t implies about the phase at time 0, assuming perfect action execution. The objective becomes:

$$\max_{\hat{\theta}_0} \sum_{t=0}^{T} \cos(\hat{\theta}_0 - \phi_t). \tag{21}$$

A.3 Closed-Form Solution

Using the trigonometric subtraction formula, we expand the summation:

$$\begin{aligned} J(\hat{\theta}_0) &= \sum_{t=0}^{T} \Big(\cos\hat{\theta}_0 \cos\phi_t + \sin\hat{\theta}_0 \sin\phi_t \Big) \\ &= \cos\hat{\theta}_0 \underbrace{\sum_{t=0}^{T} \cos\phi_t}_{C} + \sin\hat{\theta}_0 \underbrace{\sum_{t=0}^{T} \sin\phi_t}_{S}. \end{aligned} \tag{22}$$

This expression is equivalent to the dot product between the unit vector $(\cos\hat{\theta}_0, \sin\hat{\theta}_0)$ and the sum vector (C, S). The value is maximized when the vector $(\cos\hat{\theta}_0, \sin\hat{\theta}_0)$ aligns with the direction of (C, S). Therefore, the optimal initial phase is given by the angle of the vector (C, S):

$$\hat{\theta}_0^* = \text{atan2}(S, C) = \text{atan2}\left(\sum_{t=0}^{T} \sin\phi_t, \sum_{t=0}^{T} \cos\phi_t\right). \tag{23}$$

Substituting $\phi_t = \theta_t - \sum_{\tau=1}^{t} r_\tau$ back into the equation, we obtain the optimal starting phase. Finally, the optimal phase for any time step t is obtained by propagating $\hat{\theta}_0^*$ forward:

$$\hat{\theta}_t = \hat{\theta}_0^* + \sum_{\tau=1}^{t} r_\tau. \tag{24}$$

Combining these steps yields the closed-form solution presented in Eq. 7:

$$\hat{\theta}_t = \text{atan2}\left(\sum_{k=0}^{T} \sin\left(\theta_k - \sum_{\tau=1}^{k} r_\tau\right), \sum_{k=0}^{T} \cos\left(\theta_k - \sum_{\tau=1}^{k} r_\tau\right)\right) + \sum_{\tau=1}^{t} r_\tau. \tag{25}$$

B Model Architecture

Configuration	AR			MaskGIT			CMWM			
	S	M	L	S	M	L	S	M	L	XL
Embed Dim	512	768	1024	512	768	1024	384	576	768	1024
Num Heads	8	12	16	8	12	16	6	9	12	16
Hidden Dim	2048	3072	4096	2048	3072	4096	1536	2304	3072	4096
Num Blocks	8	10	12	8	10	12	8*	9*	10*	12*
# Param (M)	42	107	219	42	107	219	34	84	166	303

Table 2: Model architectures and parameter counts for different scales (Small, Medium, Large) of AR, MaskGIT, and our CMWM models. * indicates that the encoder and decoder each have this number of blocks

C Training Details

We conducted all experiments on a computational cluster equipped with 5 NVIDIA A100-SXM4-80GB GPUs. All models, including CMWM and the AR/MaskGIT baselines, were trained for a total of 100 epochs. The training duration was approximately 20 hours for each model. We employed the pre-trained VAE tokenizer from the Latent Diffusion model Rombach et al. (2022). During training, the observation sequence length was set to 128, and we used a batch size of 4 per GPU. We used the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.05. The learning rate was set to 3×10^{-5} and decayed using a Cosine Annealing scheduler with a minimum learning rate of 1×10^{-6}.

D Ablation

We conducted extensive ablation studies to validate the architectural design choices of CMWM for the prediction task (Spatial Memory and Imagination), specifically focusing on the temporal attention in the encoder and decoder (Table 3), and the configuration of the structured latent space (Table 4). All ablation experiments were performed on the robotics dataset, and we report PSNR scores for both training and test sets.

First, we evaluated the configuration of TA by toggling its application in the Encoder and Decoder and varying the attention masks (Causal vs. Bidirectional). Notably, Bidirectional Attention was excluded from the Encoder; the encoding of a current observation into a latent state must strictly depend on

present or past information to ensure the model supports initialization with observation sequences of variable lengths. In terms of performance, while introducing Bidirectional Temporal Attention in the Decoder yielded a marginal improvement in Training PSNR, the configuration without any Temporal Attention achieved the highest Test PSNR. This suggests that relying on temporal dependencies helps memorize specific training trajectories but hinders generalization to unseen environments.

Second, we observed that TPE is unnecessary. Since our model relies on the Abelian group structure where actions are commutative (i.e., the order of actions does not affect the final state), the absolute time index t is irrelevant to the state definition. The state should be determined solely by path integration, not by its timestamp.

Table 4 presents the ablation study on the hyperparameters governing the periodic structure of the latent space. This includes the ratio of "fixed" (action-invariant) neurons, the mode of coordinate division, and the period settings.

We validated the hypothesis that a portion of the latent space should encode context rather than motion. Setting the Fixed Ratio to 1/6 achieved the best balance. We compared "Global" division (partitioning the entire latent vector) against "In-token" division (partitioning within each sub-token). The Global mode significantly outperformed the In-token mode. Experimentally, we observed that the In-token configuration led to excessively high commitment loss, suggesting that fragmenting individual tokens disrupts the alignment process. This demonstrates that distributing phase properties across the global latent dimension is more effective for representing complex spatial topologies. Finally, we examined the number and distribution of periods. Following experimental findings, we set the scale ratio between adjacent periods to $\sqrt{e}$ Wei et al. (2015). This configuration not only aligns with grid data, but also constitutes the theoretically most efficient coding scheme for two-dimensional space, maximizing information capacity while minimizing representational cost. We initially utilized 16 periods, but navigation experiments revealed that using 8 periods for localization achieved the optimal accuracy. Consequently, we adopted this setting and observed a slight improvement in prediction accuracy as well. This finding suggests that a compact, multi-scale periodic representation is sufficient and more learnable than higher-dimensional or randomized alternatives.

Enc. TA		Dec. TA		TPE	PSNR	
Applied	Causality	Applied	Causality		Train	Test
-	-	-	-	-	22.17	**17.61**
✓	Causal	-	-	-	22.64	17.34
-	-	✓	Causal	-	22.63	17.50
✓	Causal	✓	Causal	-	22.28	17.42
-	-	✓	Bidirectional	-	**24.12**	17.46
✓	Causal	✓	Bidirectional	-	22.39	17.56
-	-	✓	Bidirectional	✓	23.61	17.27

Table 3: Ablation study on Temporal Attention (TA) mechanisms and Temporal Positional Encoding (TPE). We investigate the impact of applying TA to the Encoder (Enc.) and Decoder (Dec.), as well as the causality of the attention in the Decoder.

E Additional Results

We present additional quantitative comparisons, visualizations of multi-environment cognitive map learning, and a detailed analysis of our localization mechanism.

E.1 Quantitative Analysis

We provide the full quantitative results for Spatial Memory (Table 5), Spatial Imagination (Table 6) tasks and Navigation (Table 7).

Latent Settings		Period Settings		PSNR	
Fixed Ratio	Div. Mode	Number	Mode	Train	Test
0	Global	8	Fixed	22.04	17.25
1/6	Global	8	Fixed	**22.17**	**17.61**
1/3	Global	8	Fixed	21.41	17.52
2/3	Global	8	Fixed	21.17	17.55
1/6	In-token	8	Fixed	16.61	16.20
1/6	Global	16	Fixed	21.90	17.33
1/6	Global	-	Random	19.95	16.48

Table 4: Ablation study on fixed latent neurons. We investigate the impact of the Fixed Ratio, the Coordinate Division method, and Period Settings on the model performance.

Step	PSNR ↑			LPIPS ↓			FID ↓		
	AR	MaskGIT	CMWM	AR	MaskGIT	CMWM	AR	MaskGIT	CMWM
1	18.93	14.55	**23.49**	0.151	0.202	**0.102**	96.83	113.12	**86.40**
2	17.46	13.70	**23.64**	0.193	0.237	**0.100**	119.05	116.42	**81.17**
4	15.61	13.36	**23.93**	0.282	0.247	**0.095**	162.28	113.79	**78.43**
8	13.45	12.83	**24.25**	0.383	0.274	**0.091**	200.60	117.05	**73.68**
16	11.55	11.98	**24.45**	0.490	0.316	**0.088**	238.84	116.68	**76.14**
32	9.83	10.74	**24.25**	0.563	0.376	**0.086**	259.10	128.99	**74.86**
64	8.62	9.89	**24.55**	0.630	0.432	**0.083**	269.19	135.78	**70.46**
128	8.22	9.22	**23.50**	0.684	0.512	**0.111**	282.93	180.35	**89.29**
256	8.06	8.89	**23.29**	0.708	0.560	**0.118**	302.04	198.66	**96.11**

Table 5: Quantitative results on the Training set. We report PSNR (higher is better), LPIPS (lower is better), and FID (lower is better) across different prediction steps.

E.2 Multi-Map Learning

CMWM has the ability to learn multiple maps. We trained a single CMWM model jointly on the SCAND and TartanDrive datasets.

As visualized in Figure 7, the left section displays the trajectories within the global maps for both datasets, while the right section presents the corresponding predictions generated by CMWM. Experimental results demonstrate that the model can perform long-distance spatial memory, imagination, and navigation on either map using the same network weights. For this evaluation, we employed the CMWM-XL model configuration. We report the quantitative PSNR results for both datasets in Table 8.

While we used RECON and HuRoN datasets for general training in our experiments, we do not present global map visualizations or global navigation results for them. This is because these datasets do not provide unified coordinates across different trajectories. Previous world models primarily focused on short-range navigation or intra-trajectory prediction, where relative coordinates within a single clip suffice. However, our work aims to construct a global cognitive map to support long-distance navigation across the entire environment. This requires ground truth trajectories to be aligned within a unified global coordinate system, which is absent in RECON and HuRoN. In fact, a distinct map is constructed for each individual trajectory, resulting in a multitude of independent, disjoint maps within a single dataset. Therefore, while CMWM is capable of performing intra-trajectory localization on RECON and HuRoN similar to prior works, our objective of large-scale navigation inevitably entails inter-trajectory challenges that cannot be addressed due to data limitations.

E.3 Localization Mechanism

Figure 8 provides a concrete experimental visualization of the congruence solving mechanism conceptually introduced in Figure 3. It demonstrates how the model uniquely determines the physical

Step	PSNR ↑			LPIPS ↓			FID ↓		
	AR	MaskGIT	CMWM	AR	MaskGIT	CMWM	AR	MaskGIT	CMWM
1	12.49	11.89	**16.42**	0.447	0.408	**0.336**	213.13	**136.99**	217.93
2	11.65	11.52	**16.41**	0.489	0.421	**0.334**	225.88	**136.85**	226.42
4	10.81	10.80	**16.85**	0.527	0.465	**0.324**	258.11	**146.95**	216.90
8	10.13	10.07	**16.54**	0.568	0.515	**0.337**	249.57	**154.92**	224.76
16	9.55	9.68	**16.91**	0.610	0.558	**0.323**	278.24	**179.05**	218.44
32	9.32	9.33	**16.18**	0.681	0.597	**0.348**	294.97	**199.93**	238.01
64	8.82	8.55	**16.52**	0.722	0.656	**0.344**	285.96	**226.70**	229.87
128	8.10	8.58	**16.98**	0.762	0.650	**0.329**	316.96	252.36	**219.49**
256	8.16	8.14	**16.51**	0.778	0.679	**0.342**	327.07	272.98	**237.25**

Table 6: Quantitative results on the Test set. Comparison of AR, MaskGIT, and CMWM models.

		Train			Test		
		CMWM	MaskGIT	AR	CMWM	MaskGIT	AR
AE (↓)	Short-range	$\mathbf{0.18}_{\pm 0.11}$	$0.45_{\pm 0.21}$	$0.69_{\pm 0.39}$	$\mathbf{0.45}_{\pm 0.10}$	$0.89_{\pm 0.47}$	$1.20_{\pm 0.68}$
	Long-range	$0.20_{\pm 0.08}$	-	-	$0.48_{\pm 0.28}$	-	-

Table 7: Comparison of model navigation performance across different ranges. AE represents the absolute error between the ground truth and predictions (in meters).

displacement d by aggregating phase information, covering both (A) short-range and (B) long-range localization scenarios.

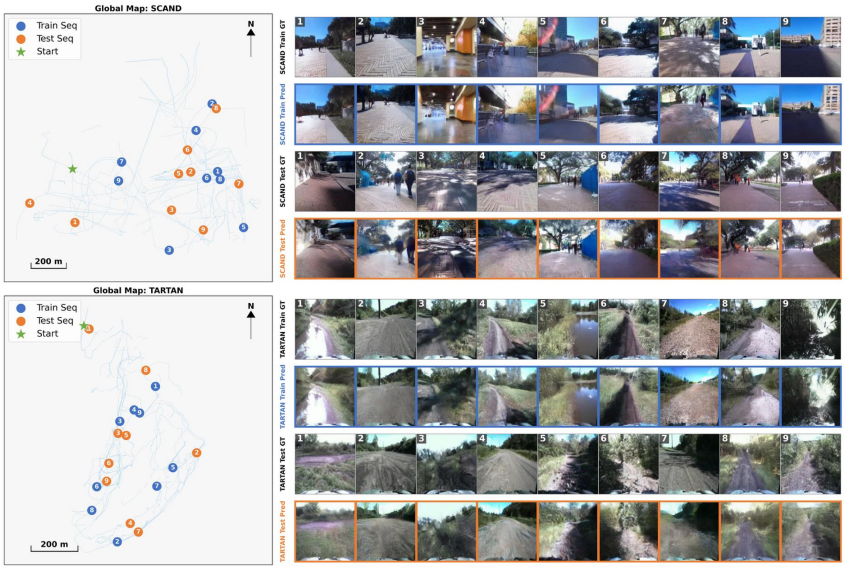

Figure 7: Multi-Map Learning with a Unified Model. We demonstrate that a single model is trained simultaneously on both datasets. The left panels display the global map trajectories for the SCAND and TartanDrive datasets, while the right panels present ground truth observations alongside the corresponding CMWM predictions.

Dataset	SCAND		TartanDrive	
	Train	Test	Train	Test
PSNR	20.80	17.49	21.91	18.52

Table 8: Quantitative results for Multi-Map Learning on SCAND and TartanDrive datasets.

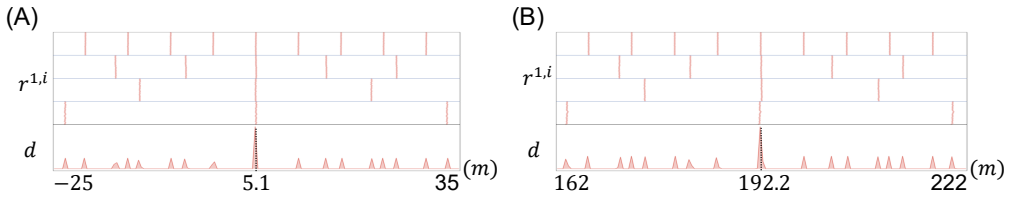

Figure 8: Resolving the true physical displacement d using eight different periods (only four are shown for clarity). Due to periodicity, single-module differences yield multiple candidate values. By aggregating candidates across four incommensurate periods, the true displacement is uniquely identified as the value where candidates coincide.