Building Better Multi-Agent Systems: the Theory and Practice of Multi-Agent World Models

Mark Cavolowsky

May 8, 2025

Abstract

From coordinating robotic warehouses to managing autonomous vehicle fleets, agents in a Multi-Agent System (MAS) must understand and predict the interactions of multiple agents within the world. While feature-engineered World Models (WMs) perform well in controlled environments and learned WMs show promise in single-agent scenarios, both struggle with complex, high-dimensional multi-agent dynamics. This survey introduces a framework to help practitioners select and implement Multi-Agent World Model (MAWM) approaches based on their specific multi-agent needs. It provides actionable guidelines while identifying key research challenges in scaling, distributed consistency, and task-agnostic transfer. The work synthesizes recent advances in MAWMs offering a roadmap for future research in multi-agent systems.

Contents

1 Introduction

Multi-Agent Systems (MASs) power numerous applications from autonomous vehicle fleets [47] to multi-robot manipulation [12]. These applications rely on accurate World Models (WMs) [118] to predict and manage the interactions

between agents, but traditional feature-engineered approaches are limited by human knowledge. Richard Sutton's Bitter Lesson [104] teaches readers two things: (1) the world is intractably complex and (2) feature engineering will inevitably lose to the computational scaling of learning and search. This necessitates the creation of learned WMs to perform in environments with complex, coupled dynamics, where agents must predict and adapt to each other's actions [45]. Recent advancements in single-agent learned WMs [35, 37, 38] offer a compelling alternative by leveraging data-driven techniques to model these interactions more effectively. However, adapting these single-agent techniques to MASs requires the development of Multi-Agent World Models (MAWMs) to develop learning objectives and architectures that account for multiple WMs communicating across a network to collectively solve problems. This survey is aimed at researchers and practitioners in MASs, particularly those tackling environments with complex, dynamic interactions and looking for improvements in sample efficiency, task performance, generalization, and more. Readers will gain an in-depth understanding of the latest approaches in learned WMs, practical guidelines for selecting and implementing these models, and insights into key trade-offs and open challenges in the field.

To motivate the need for multi-agent world models, consider first a simple scenario: two quadcopters must coordinate to move a single, heavy payload from point A to point B. Even in this setup, the system exhibits complex dynamics: quadcopter downwash affects neighboring vehicles, communication delays introduce information asymmetries, and joint lift requires precise synchronization to maintain payload stability. Traditional methods such as single-agent models or hand-coded swarming rules [92] struggle because they require explicit modeling of these effects, which is intractable in real-world conditions. Scaling this up to a disaster-response setting—where a fleet of quadcopters must locate survivors, navigate unstable terrain, and transport medical supplies—further exposes the complexity. In addition to delicate aerodynamic couplings and tight joint state prediction, each agent sees only local regions of a cluttered environment, and damaged infrastructure severely limits communication, making centralized control difficult or impossible. While one might attempt to address these multi-agent dynamics by painstakingly engineering features—for example, rules encoding how propeller wash or payload configurations affect neighbors—this quickly becomes infeasible in the face of changing conditions, partial observability, and emergent behaviors. Learned world models offer a more powerful alternative by discovering implicit couplings directly from data and adapting to new situations without constant human intervention. They leverage latent representations to capture complex dependencies among agents, enabling robust performance under partial information and limited connectivity. Although distributed control strategies remain essential for certain aspects of coordination, their reliance on well-defined system equations often falls short in disaster scenarios characterized by uncertainty and rapid change, making learned world models a crucial tool for operating effectively in these challenging environments.

This survey delivers the first in-depth analysis of learned WMs for MASs, presenting an analytical framework that clarifies when and how to use MAWMs effectively. Through an exploration of the current approaches, it defines the fundamental trade-offs in multi-agent world modeling, showing how choices in WM design can influence parameters such as scalability, communication efficiency, and coordination effectiveness. The survey also identifies critical open challenges, such as scaling models to handle large groups of agents, building task-agnostic representations, maintaining consistency in distributed settings, and providing agents with formal guarantees. Together, these insights aim to bridge the gap between research advancements and real-world implementation.

The rest of the paper is organized to guide readers through these ideas. It begins with background on classical and single-agent WMs, setting the stage for understanding how MAWMs differ (Section 2). MAWMs are introduced next, along with an analysis framework, which examines various architectural approaches, learning objectives, and real-world applications of MAWMs (Section 3). The paper teases apart practical insights into the costs and benefits of different design choices (Section 4). Finally, it concludes by identifying open challenges and offering directions for future research, with the goal of advancing scalable and reliable MAWMs for complex, real-world systems (Section 5 and Section 6). The appendices contain supplementary material including the survey methodology (Appendix A.1), related surveys (Appendix A.2), a generative AI usage statement (Appendix A.3), and acronym definitions (Appendix B).

2 World Modeling Background

Prior to examining MAWMs, it is crucial to understand the foundations of world modeling in general. A *World Model (WM)* is a representation of an environment that allows agents to simulate interactions and dynamics to predict and optimize their behaviors. To do this, WMs must perform two primary functions: 1) learning tractable representations of system state from observations and 2) predicting the evolution of these representations over time. These functions are generally split over three major components: an encoder module, a transition module, and a decoder module. The *encoder* module maps a history of agent sensors and observations to world states (e.g., Simultaneous Localization

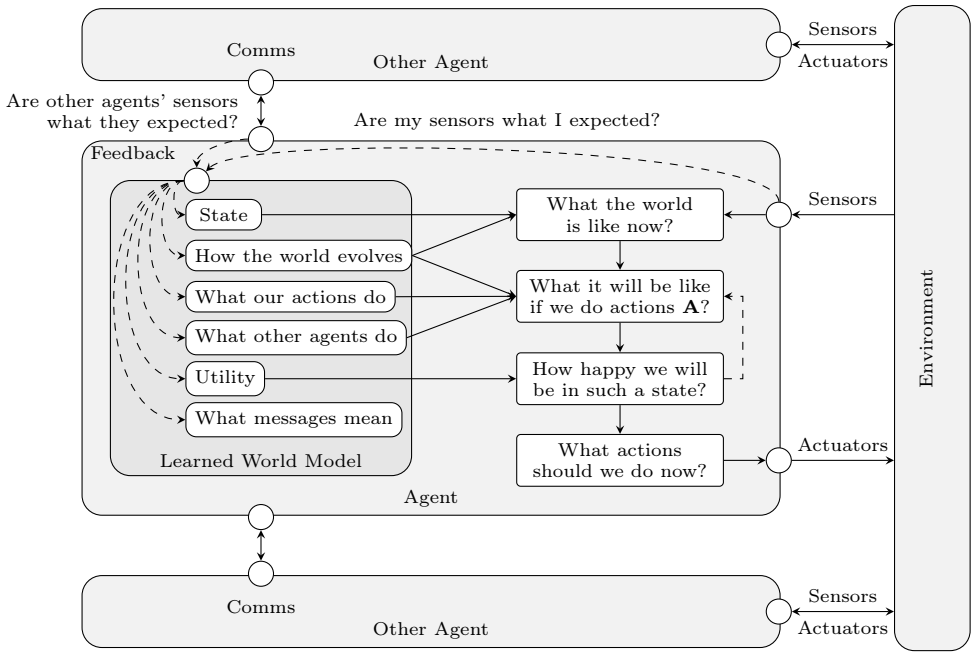

Figure 1: Architecture of a model-based utility agent with a learned multi-agent world model. The diagram shows: (1) a learned world model (left) that maintains internal representations of the environment's state and dynamics, (2) a communication system that coordinates with other agents, and (3) an execution system (right) that uses this model for decision-making. The world model contains the current state estimate, how the environment evolves, the effects of agents' actions, and utility preferences. The execution system processes this knowledge through: (1) understanding the current world state, (2) predicting future states after potential actions, (3) evaluating happiness (utility) in those predicted states, and (4) selecting optimal actions. Two critical feedback loops enable continuous learning and adaptation: (1) a learning loop that validates whether sensory inputs match the model's predictions, and (2) a planning feedback loop that refines actions based on expected outcomes. The agent connects to its environment through sensors and actuators and to other agents through communications, forming a closed-loop control system. This architecture enables the agent to simultaneously learn from experience, evaluate plans and actions using its learned model, and adapt its behavior based on prediction accuracy and achieved utility. Image adapted from Russell and Norvig [94].

and Mapping (SLAM) camera histories to point clouds [138], observations to latent histories in Recurrent Neural Networks (RNNs) [44]). The *transition* module takes the encoded states and uses an action model to predict the evolution of states over time. The *decoder* module then takes these predicted states and projects agent outputs (e.g., observations or rewards).

While expert-designed models have been used for decades—even before the advent of computers—recent advancements in learning algorithms have shifted the focus toward agents that can *learn* these representations from data [35, 37, 116]. Real-world environments are inherently complex and unpredictable [104], and agents are limited by their finite computational resources. Learned WMs help bridge this gap by enabling agents to reason about and predict their surroundings—making learning, decision-making, and problem-solving more efficient. Understanding these foundations is essential for exploring how MASs can leverage learned WMs to address even greater challenges.

As shown in Fig. 2, world modeling research spans two key dimensions—agent count (single vs. multi) and representation type (feature-engineered vs. latent)—creating four distinct "quadrants":

1. Single-agent feature-engineered models that leverage expert knowledge (Section 2.1);
2. Multi-agent feature-engineered models that extend expert structures to handle multiple interacting agents (Section 2.2);
3. Single-agent latent models that learn representations without assumptions (Section 2.3); and
4. Multi-agent latent models - the focus of this survey and covered in subsequent sections.

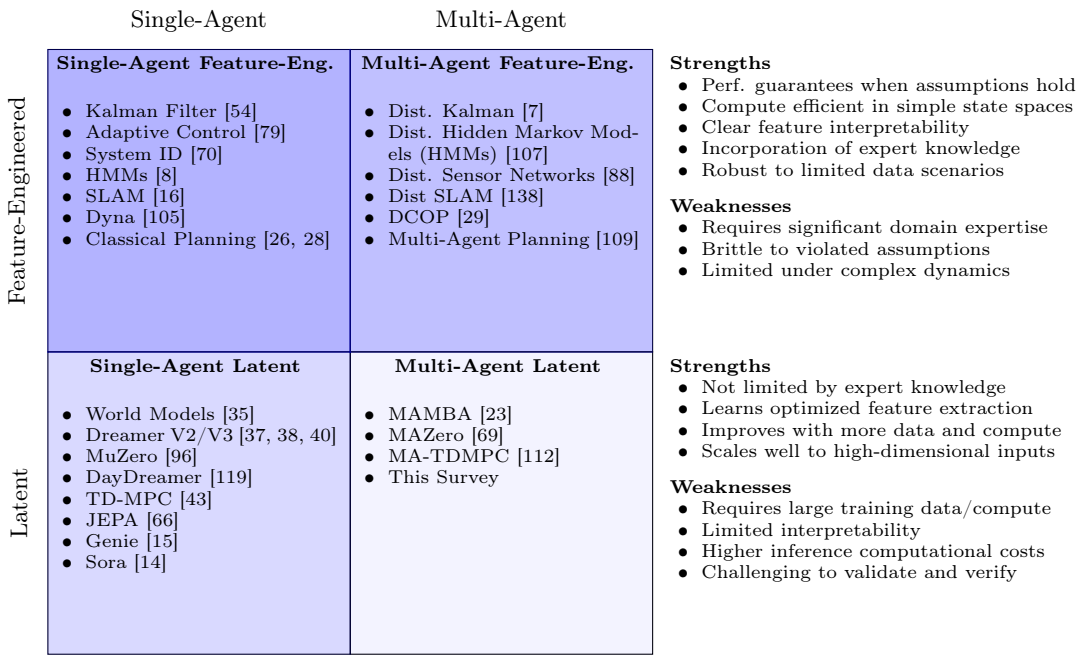

Figure 2: Research landscape across world modeling approaches. The quadrants show both research density (blue shading) and key works in each category. While single-agent approaches and feature-engineered multi-agent approaches have extensive literature, multi-agent latent world models remain relatively unexplored—motivating this survey.

The first three quadrants provide essential context and foundations for understanding modern latent MAWMs, which this section examines prior to the remainder of the paper on learned MAWMs.

2.1 Single-Agent Feature-Engineered Models

Single-agent feature-engineered models represent an early cornerstone in the development of WMs, with roots tracing back to control theory and expert-designed systems. These models use domain knowledge to craft representations tailored to particular tasks, enabling accurate predictions, decision-making, and control. In feature-engineered approaches, the encoder, decoder, and transition model are all expert-designed based on an *a priori* understanding of the problem. The encoder typically maps raw observations to predefined state variables (e.g., pose, velocity, world geometry) based on domain knowledge. The decoder reconstructs observations from these state variables using inverse transformations or sensor measurement models. Depending on the domain, the transition model often takes the form of either discrete or continuous differential equations describing system dynamics, such as linear state-space models for simple systems or nonlinear equations for more complex dynamics. Early approaches often relied on handcrafted mathematical descriptions of systems, and over time, advancements in control theory and artificial intelligence have integrated expert insights with data-driven techniques to enhance effectiveness.

State-Space Models The Kalman filter [54] represents a seminal contribution to state-space WMs. It focuses on estimating system states from noisy sensor measurements and a state-action transition model, and its variants still endure to this day (e.g., [56]). HMMs [8] are important approaches that provide a probabilistic framework for modeling sequential data with unobserved (hidden) states that generate observable outputs. While powerful for sequence modeling, traditional HMMs learn the hidden states but require a known transition and observation model. SLAM [16] represents a foundational approach to geometric world modeling that enables agents to construct and maintain spatial representations of unknown environments while simultaneously tracking their position within them. Classical SLAM systems integrate multiple control-theoretic elements: Kalman filtering for state estimation, probabilistic models to handle sensor and motion uncertainties, and optimization techniques like bundle adjustment or pose graph optimization to maintain global consistency. The success of SLAM in robotics and autonomous systems demonstrates both the power of feature-engineered WMs and their limitations—while effective for geometric mapping and localization, they typically rely on hand-crafted features and environment assumptions that may not

generalize to more complex scenarios. While many of the above approaches provide provable guarantees [53], they have limitations. Many assume highly abstracted models with simplifying assumptions that require a good inductive bias (e.g., the Kalman filter assumes linearity and Gaussian noise; HMMs require a known state, transition, and observation model, etc.), limiting their applicability to well-characterized problems (e.g., low-dimensional and known system models).

Symbolic World Models Single-agent symbolic WMs form a cornerstone of classical artificial intelligence, providing structured representations of environments through discrete symbols and logical relationships. These models typically employ formal languages like Stanford Research Institute Problem Solver (STRIPS) [28] or Planning Domain Definition Language (PDDL) [1] to encode states, actions, and transitions, enabling agents to reason about their environment through symbolic manipulation. Building on classical planning foundations, approaches like Hierarchical Task Network (HTN) decompose complex goals into manageable subtasks, offering scalable solutions [26]. While these symbolic approaches excel in domains with clear rules and deterministic outcomes, they often face challenges in handling uncertainty and continuous state spaces, leading to hybrid approaches that combine symbolic reasoning with probabilistic methods [52].

Data-Driven Feature-Engineering Approaches As research in Artificial Intelligence (AI) progressed, learned WMs were developed to tackle increasingly complex, high-dimensional problems through data-driven approaches. One foundational contribution is Dyna [105], which proposed an integrated framework for planning, learning, and reactive execution in Reinforcement Learning (RL). Dyna used a learned state-action transition model to generate "hypothetical" experience, allowing agents to simulate rollouts and improve sample efficiency through planning. Building on control-theoretic foundations, advancements in adaptive control [79] and system identification [70] have further improved predictions of system behaviors and have enabled more effective decision-making and control strategies in complex, dynamic environments. Probabilistic Inference for Learning COntrol (PILCO) [20] uses a Gaussian Process (GP) model to learn system dynamics from data. This synthetic data model aids in training a control policy and the authors demonstrate its effectiveness on control tasks (e.g., inverted pendulum and cart-pole). However, while effective for simpler, low-dimensional tasks, PILCO struggles with the challenges of scaling to high-dimensional problems due to the cubic computational complexity of GPs ($O(n^3)$ with respect to the number of dimensions. However, these approaches' reliance on explicit state representations prove limiting in high-dimensional, continuous environments. These limitations underscore the need for more scalable, flexible world modeling, and pave the way for more modern approaches.

2.2 Multi-Agent Feature-Engineered Models

MASs predominantly rely on multi-agent adaptations of *feature-engineered* models that encode expert-specified domain knowledge into explicit state representations and transitions. Multi-agent feature-engineered models are a natural extension of single agent approaches, but multiple agents increase the complexity significantly, increasing the probability that domain experts introduce biases, limiting assumptions, or oversimplifications. Similar to single-agent approaches, the encoder, decoder, and transition models in multi-agent feature-engineered WMs are based on an *a priori* understanding of the problem. The encoder must incorporate information from additional agents into the joint encoded state, and the decoder reconstructs either global observations or local observations across agents. Finally, the transition model must predict the evolution of the joint state given the joint action across the agents. Understanding these classical approaches—along with their inherent scalability and adaptability limitations—is crucial to motivating the transition toward learned MAWMs capable of discovering structure from rich, high-dimensional data.

State-Space Models Multi-agent feature-engineered models enable the calculation of consistent global states from local observations. The distributed Kalman filter [84] extends single-agent optimal state estimation to multi-agent systems, enabling each agent to maintain a local WM while sharing information with neighbors. Early approaches explored sharing state estimates [84], sensor data and covariance [84], and hierarchical communication architectures [67]. The consensus extended Kalman filter [7] further develops this approach for nonlinear systems, providing theoretical stability guarantees under network connectivity and collective observability conditions. Distributed HMMs provide another approach to decentralized state estimation. Each agent maintains a local probabilistic model and exchanges messages over communication graphs to form unified state estimates. These systems employ various information fusion strategies: exact bookkeeping for precise but computationally expensive tracking, conservative fusion

for reliability at the cost of performance, or hybrid approaches like Tamjidi et al.'s integration of Iterative Conservative Fusion (ICF) with consensus methods [107]. Multi-agent SLAM systems exemplify how feature engineering leverages geometric constraints and sensor characteristics to build joint maps of the environment [138]. Classical multi-robot SLAM factorizes the joint state into robot poses and landmark positions, applying optimization tools like bundle adjustment or incremental smoothing to ensure global consistency. Working together in multi-agent SLAM, agents can perform collaborative pose estimation, collaborative mapping, loop closure, feature extraction and matching, outlier rejection, and system initialization. Despite their effectiveness in structured domains, these approaches face fundamental challenges in scaling to large teams and dynamic environments. Their reliance on expert-defined models limits their ability to capture complex real-world dynamics, particularly in scenarios heterogeneous sensors or complex inter-agent interactions. Finally, Peddemors and Yoneki [88] proposed the integration of the above concepts on a large scale through a distributed WM using probabilistic methods and cooperative sensing, where agents maintain and share local data through peer-to-peer communication while validating against false information. The framework employs compositional hierarchies to fuse multi-sensor data in a bottom-up manner, identifying frequent spatio-temporal patterns to generate higher-level, symbolic knowledge from elementary sensor information.

Symbolic World Models Symbolic approaches to multi-agent world modeling have seen significant development in cooperative multi-agent planning. A key framework is MA-STRIPS [13], which provides a minimalistic multi-agent extension of the STRIPS planning model that enables modeling private and public information between agents. Building on this, several approaches have emerged including hierarchical methods like HTN planning for coordinating agent teams [18, 106] and distributed versions of classical planners [80]. These symbolic frameworks enable representing both the individual capabilities of agents and their shared knowledge and coordination requirements. For example, in cooperative domains, HTN approaches leverage abstraction levels to enhance efficiency in coordinating agents' plans [106]. Recent work has focused on developing distributed heuristic functions that can guide multi-agent search [101, 139], enabling better coordination during plan generation. Additionally, other approaches have focused on developing robust communication protocols [83], allowing them to share critical information during planning while managing computational overhead. The symbolic nature of these models enables formal analysis of properties like completeness [108], optimality [82], and privacy preservation [140]. However, these symbolic approaches face significant limitations in complex real-world settings—they generally assume perfect abstractions, struggle with temporal reasoning, and have difficulty scaling agents in real-time cooperative planning in dynamic environments [109]

2.3 Single-Agent Latent Models

While feature-engineering methods offered strong theoretical foundations, the emergence of deep learning enabled simultaneous learning of model encoders, transition models, and decoders from high-dimensional data. In contrast to approaches that identify parameters within a fixed model structure, latent models focus on learning relevant representations of the world. The encoder maps observations into a learned latent space that captures relevant features automatically, often using Neural Networks (NNs) to compress high-dimensional inputs into compact vector representations. The decoder learns to reconstruct observations from these latent variables, discovering an efficient compression of the essential dynamics. The transition model operates entirely in this learned space, predicting how latent states evolve over time without requiring explicit physical equations or domain knowledge. This data-driven approach enables latent models to handle complex, high-dimensional environments where manually designing models would be impractical.

Koopman Models An early latent model was the Koopman operator [61]: a transformation that maps a low-dimensional, non-linear problem into a infinite-dimensional, linear space. This transformation allows the application of linear control techniques, which are well understood and computationally efficient, to otherwise intractable nonlinear systems. Classical Koopman approaches have been successfully applied to problems in fluid dynamics, robotics, and control [73, 74], where identifying global, linear representations of complex dynamics enables predictive modeling and controller design. Modern extensions of Koopman theory incorporate deep learning to handle high-dimensional data and expand its applicability. NNs are often employed to learn mappings (i.e., "lifting functions") that embed nonlinear dynamics into approximate finite-dimensional linear spaces. For instance, Li et al. [68] proposed a method combining deep NNs with Koopman operators, demonstrating success in applications such as flexible objects and swimming robots.

Early Latent Models With the early successes of deep learning, authors began applying that to the creation of learned latent WMs for improved RL and optimal control. Embed to Control (E2C) [116] addresses the challenge of mapping high-dimensional worlds to a reduced-order model through a Variational AutoEncoder (VAE) framework that learns image-based reconstructions from forward latent predictions. E2C introduced two key innovations: (1) a VAE architecture that explicitly constrains the latent space to be locally linear for control, and (2) a Kullback-Leibler (KL) divergence term that enforces consistency between predicted and encoded states, enabling stable long-term predictions. The model demonstrated strong empirical performance on visual control tasks including pendulum swing-up and cart-pole balancing, both establishing an important benchmark for and inspiring subsequent works (e.g., PlaNet [36] and Dreamer [37]). While similar to E2C, Stochastic Optimal control with LAtent Representations (SOLAR) differs from other latent WMs by jointly optimizing its representation to make local linear-Gaussian dynamics models more accurate, rather than focusing on global reconstruction or forward prediction. It employs a deep Bayesian Linear Dynamical System (LDS) model with a probabilistic graphical model structure to infer dynamics from data. The model maintains both global dynamics priors and local time-varying linear-Gaussian dynamics, enabling efficient policy improvement through Linear Quadratic Regulator (LQR) while handling partial observability through latent state estimation.

Factorized World Models Recent latent WMs employ various state factorizations to capture different aspects of environment dynamics. By separating representations into meaningful components like deterministic and stochastic elements [36] or controllable and uncontrollable dynamics [85], these factorizations provide inductive biases that help models learn more accurately and with fewer samples. First, Hafner et al. [36] introduces a WM based on the Recurrent State Space Model (RSSM), which factors the latent space into deterministic and stochastic components:

$$\ell_t = (h_t, z_t) \qquad \text{(latent decomposition)} \tag{1}$$

$$h_{t+1} = f_\theta(h_t, z_t, \mathbf{a}_t) \qquad \text{(deterministic transition)} \tag{2}$$

$$z_{t+1} \sim p_\theta(z_{t+1}|h_{t+1}) \qquad \text{(stochastic transition)} \tag{3}$$

This separation allows deep Planning Network (PlaNet) to capture both deterministic system dynamics and stochastic environmental factors in the transition model. Using the Cross Entropy Method (CEM) for efficient planning in latent space and a novel latent overshooting objective, PlaNet achieved 200× better sample efficiency than contemporary model-free approaches on complex continuous control tasks.

Dreamer [37] presents an actor-critic RL agent that learns behaviors entirely by propagating gradients through an RSSM-based WM. Dreamer directly learns both a policy network and a value network in latent space, optimized through backpropagation of these value estimates through the WM dynamics. This enables efficient credit assignment across long horizons while being more computationally tractable than per-timestep CEM optimization. Significant extensions to the Dreamer architecture progressed from DreamerV2's discrete latent variables [38] and DreamerV3's robustness techniques [40] to Director's hierarchical policies [39]. These iterations ultimately enable DayDreamer's sample-efficient real-world robotic learning of tasks like quadruped locomotion in one hour [119]. Recent work by Sun et al. [103] highlights that traditional WMs, such as those used in the Dreamer series [37, 38, 40], often struggle with visual pixel-based inputs containing exogenous or irrelevant noise. Their Hybrid Recurrent State Space Model (HRSSM) [103] combines a masking strategy with a bisimulation principle to capture task-relevant features while filtering out irrelevant spatio-temporal details, learning expressive representations in noisy environments.

Building on the base factorization of RSSMs, Pan et al. [85] introduces Iso-Dream, which separates the model into a three-branch architecture that explicitly separates controllable (ego agent) and non-controllable (other agents) dynamics through inverse dynamics learning. This separation allows the model to handle complex multi-agent scenarios like autonomous driving with 30 vehicles without requiring explicit communication, instead learning to predict other agents' behaviors through the non-controllable branch. The approach demonstrates that explicit factorization of agent dynamics can improve both prediction accuracy and control performance. Kipf et al. [59] introduce a different factorization approach with Contrastively-trained Structured World Models (C-SWMs), which learn object-oriented latent representations and transitions without pixel-based reconstruction. C-SWMs utilize a contrastive approach for representation learning in environments with compositional structure. The model structures each state embedding as a set of object representations and their relations, modeled by a Graph Neural Network (GNN) [135], allowing objects to be discovered from raw pixel observations without direct supervision as part of the learning process. The contrastive learning objective enables C-SWMs to focus on task-relevant features while discarding irrelevant visual details, addressing limitations of reconstruction-based losses.

World Models for Planning While WMs demonstrate the promise of zero-shot adaptation to new problems with online planning, planning in a learned latent space can create error stacking problems [77, 120], requiring WMs designed for online planning. MuZero [96] integrates a learned latent model with Monte-Carlo Tree Search (MCTS) [99] to predict only planning-critical quantities (policies, values, rewards), achieving state-of-the-art results across both board games and Atari environments. Another latent planning approach, Learning Latent Landmarks for Planning (L^3P) [128] leverages a WM to learn and plan latent landmarks in goal space using an A* algorithm, improving temporally extended reasoning. L^3P embeds observations and forms graph nodes through clustering with edges representing approximate costs between goals, allowing compact latent-space planning through a higher-level abstraction. Temporal Difference Model Predictive Control (TD-MPC) [43] employs a learned, task-oriented latent dynamics model for short-horizon Model-Predictive Control (MPC)-based trajectory optimization and a terminal value function for long-term return estimation. Both models are trained jointly through temporal difference learning, providing a reward-guided feature space and avoiding the reconstruction of state details. Building on this, Temporal Difference Model Predictive Control 2 (TD-MPC2) [42] learns versatile WMs from large, uncurated datasets across multiple domains, introducing architectural improvements (e.g., LayerNorm, Mish activations, SimNorm, discrete regression, and task embeddings) for improved stability and scaling up to 317M parameters on 80+ tasks. Finally, Joint Embedded Predictive Architecture (JEPA) [66] is a broad intelligence framework including an energy-based model for differentiable online planning, enabling both reactive and deliberative behaviors seamlessly.

Generative AI World Models Recent breakthroughs in Generative AI (GenAI) demonstrate increasingly sophisticated world understanding of physical dynamics and causal relationships. Imagination with auto-Regression over an Inner Speech (IRIS) [75] achieves human-level performance on the Atari 100k benchmark in just two hours of gameplay using a WM with a discrete autoencoder and autoregressive Transformer. Genie [15] advances environment generation by learning to create interactive 2D worlds from text or image prompts through unsupervised learning from internet videos. VideoPoet [60] integrates Large Language Models (LLMs) with masked reconstruction, enabling improved temporal coherence and physical understanding in generated scenarios. In autonomous driving, Generative AI for Autonomy-1 (GAIA-1) combines scene understanding with video diffusion for sensor reconstruction. With 6.5B parameters trained on 4,700 hours of driving, it exhibits scaling laws similar to LLMs. DINO World Model (DINO-WM) introduces zero-shot visual planning through a pre-trained Vision Transformer (ViT) architecture for embedding prediction and enabling goal-directed optimization without expert demonstrations. Sora [14], a text-guided video generation model, demonstrates sophisticated reproduction of complex dynamics through physically consistent videos, showing a strong grasp of object permanence and cause-effect relationships. While these developments showcase the potential of language-enhanced compositional reasoning and emergent capabilities at scale, challenges remain in evaluating physical understanding versus pattern matching [66] and addressing video quality constraints [15].

3 Multi-Agent World Models (MAWMs) Framework

A Multi-Agent World Model (MAWM) is a multi-agent extension to the single-agent latent WM that enables agents to reason about, predict, and coordinate in complex environments by maintaining distributed representations of the world state. Unlike single-agent WMs that abstract multi-agent effects into environmental non-stationarity/stochasticity [11, 85], MAWMs explicitly factor the impacts on the learned world state into individual agent contributions. This factorization improves multi-agent planning and acting through enhanced credit assignment and improved online policy optimization. Similar to their single-agent counterparts, a MAWM is broadly broken down into encoder, transition, and decoder modules. The encoder maps all agent observations and communications into a learned latent space that captures relevant features, and the decoder learns to reconstruct observations from these latent variables. The transition model predicts the evolution of the latent variables as a function of the joint action of all of the agents. MAWMs are diverse in both structure and application, addressing challenges such as compressed joint state representations, coupled transition dynamics, and extra-agentic action prediction.

Definition 3.1 (MAWM)**.** *A MAWM for N agents is defined as a tuple $\mathcal{M} = \langle \{\mathcal{M}^i\}_{i \in K}, G, \mathcal{C} \rangle$, where:*

- $\{\mathcal{M}^i\}_{i \in K}$*: Set of K local world models where $1 \leq K \leq N$ each comprising:*
  - Ω^i*: Local observation space;*
  - $\mathcal{A}^i$*: Local action space;*
  - $\mathcal{L}^i$*: Local latent state space;*

- $\mathcal{E}^i : \Omega^i \times \tau^i \to P(\mathcal{L}^i)$*: Encoder function mapping observations and history* ($\tau^i = \prod_t o_t^i$) *to distributions over latent states;*
- $\mathcal{D}^i : \mathcal{L}^i \to P(\Omega^i)$*: Decoder function mapping latent states to distributions over observations;*
- $\mathcal{T}^i : \mathcal{L}^i \times \mathcal{A}^i \to P(\mathcal{L}^i)$*: Transition function predicting the next latent state distribution given the current latent state, action, and received messages; and*
- $\mathcal{R}^i : \mathcal{L}^i \times \mathcal{A}^i \to \mathbb{R}$*: Reward function predicting rewards given latent states and actions;*

- $G(V, E)$*: Communication graph defining the information flow; and*
- $\mathcal{C} : \prod_{i\in\mathcal{I}} \mathcal{L}^i \times G \to \prod_{i\in\mathcal{I}} \mathcal{L}^i$*: Communication function that updates agent latent states.*

Special cases include:

- ***Centralized Multi-Agent World Model (C-MAWM):*** *Single model with centralized communication* ($E = \{(i, m), (m, i) \mid i \in \mathcal{I}\}$)*;*
- ***Decentralized Multi-Agent World Model (D-MAWM) — Full Communication:*** *All-to-all connectivity* ($E = V \times V \setminus \{(i, i) \mid i \in V\}$)*; and*
- ***D-MAWM — No Communication:*** *Independent agents* ($E = \emptyset$)

with the generic D-MAWM with Graph Communication generalizing all of the above:

- ***D-MAWM — Graph Communication:*** *Arbitrary topology* ($E \subseteq V \times V$)*.*

A systematic analysis of MAWMs requires the examination of three fundamental aspects: (1) architectural design and communication, (2) learning objectives, and (3) functional applications in MASs. The architectural design (Section 3.1) determines how WMs are distributed across agents and how information flows between them. Learning objectives (Section 3.2) shape how these models acquire and maintain representations of the environment and agent interactions. Finally, the functional applications (Section 3.3) describe how these WMs enhance various aspects of MAS performance, from reducing environmental sampling to enabling online planning and control.

3.1 Architectures for MAWMs

MAWMs have diverse communication architectures—including both centralized and decentralized approaches—which define WM structure and the information flow. C-MAWMs employ a singular world model, where each agent communicates with the C-MAWM for predictions and updates. In contrast, D-MAWMs distribute multiple WMs across agents, enabling local estimations and reducing dependence on a centralized controller. Distributed architectures can be refined by their communication paradigms: no, all-to-all, and graph-based communication topologies. No communication avoids failure modes due to unreliable or adversarial communication regimes, but they require computation on the agent to perform implicit communication through sensing and Opponent Modeling (OM). All-to-all communication ensures that each agent has all information to make decisions, but it maximizes bandwidth costs and eliminates situations where agents communications are range-limited. Finally, graph-based communication regimes are the most general, encompassing both no communication and all-to-all communication regimes in specific graph topologies, while being sufficiently expressive to account for more complex cases across arbitrary networks. It is important to note that some approaches, particularly those employing the Centralized Training, Decentralized Execution (CTDE) framework in Multi-Agent Reinforcement Learning (MARL), blur the distinction between centralized and decentralized architectures. In such cases, this work does its best to categorize the work based on where the *world model* is used (for example, a CTDE training WM would be centralized [17], but a CTDE planning WM would be decentralized [111]), understanding that no categorization framework perfectly encompasses all approaches. The following subsections analyze these architectural approaches in detail, beginning with centralized MAWMs before examining decentralized variants with increasing levels of communication complexity.

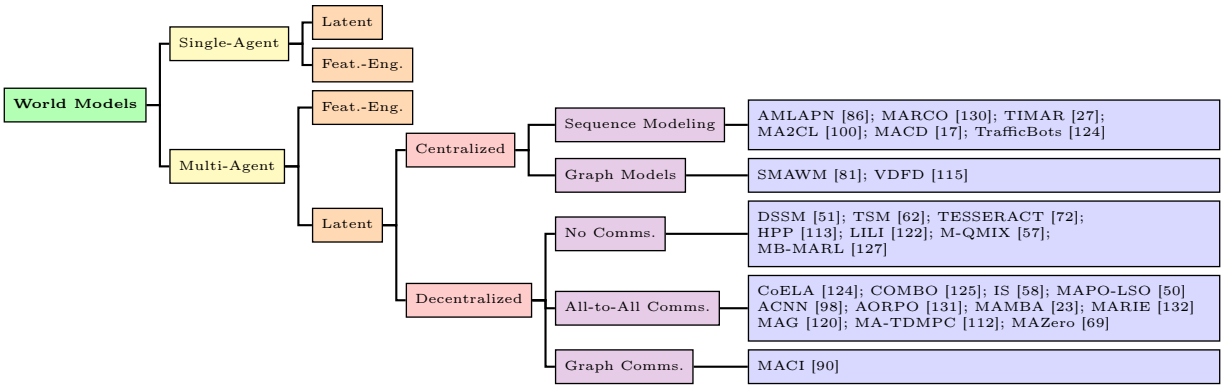

Figure 3: Taxonomy of World Model Research. The hierarchy organizes approaches by their fundamental architectural choices. The top two levels represent the four quadrants: single-agent feature-enginered models, multi-agent feature-engineered models, single-agent latent models, and multi-agent latent models. The tree teases apart the multi-agent latent models branch into decentralized architectures with varying communication patterns (graph-based, all-to-all, or none) and centralized approaches using either graph models or sequence modeling. Each leaf node lists key papers in that category.

3.1.1 Centralized Multi-Agent World Models (C-MAWMs)

Centralized Multi-Agent World Models (C-MAWMs) are a natural extension of single-agent WMs by treating multiple agents as components of a unified system state, where a single, centralized model captures the dynamics of all agents and their interactions through a singular joint state representation [5, 17, 27, 41, 81, 86, 97, 100, 102, 115, 117, 123, 130]. The centralized architecture enables direct access to all agent observations and actions for prediction and planning, eliminating communication overhead between models. However, centralized computation scales exponentially with the number of agents due to the curse of dimensionality in joint action and state spaces [45], with complexity $O(|A|^N)$ in the joint action space alone. This exponential scaling makes naive C-MAWMs computationally intractable for large agent populations. Additionally, the centralized architecture creates a single point of failure, making it ill-suited for scenarios requiring decentralized execution or fault tolerance.

Recent work has focused on making centralized approaches more tractable through various approximations and structural constraints. Multi-Agent RL with Centralized mOdels and exploration (MARCO) [130] addresses the NEXP-complete complexity of Decentralized Partially Observable Markov Decision Processs (Dec-POMDPs) [10] through a centralized approximate model achieving $O(\text{poly}(|S|, |A|))$ sample complexity. By learning a single stationary model that generalizes across policies, it avoids exploring the exponential joint-policy space. The approach alternates between model learning and policy optimization within the learned model, while employing a separate exploration policy trained to maximize both environmental reward and model uncertainty reduction. This targeted exploration improves data efficiency compared to traditional Dyna-style approaches that rely solely on the current policy.

Sequence Modeling Approaches Most authors using centralized models provide the WM full access to each agent's information in the form of a joint observation, action, and/or state. Given the fully-centralized architecture, many authors take advantage of sequence modeling approaches such as RNNs [30] and Transformers [110] to learn the optimal communication features to attend to at any given time [27, 86, 100, 124, 132]. For example, Park et al. [86] propose a model-based MARL method for competitive games using RNN-based actor-critic networks and deterministic policy gradients. To address non-stationarity due to evolving agents, they introduce a pseudo-WM to learn auxiliary prediction networks for modeling state transitions, reward functions, and opponent behavior. Their approach combines CTDE through recurrent layers that enable differentiable communication between agents. The auxiliary networks promote understanding of environment dynamics and opponent behavior while maintaining model-free efficiency. Empirical results demonstrate improved stability and performance compared to model-free approaches across various competitive scenarios. Alternatively Feng et al. [27] and Song et al. [100] handle agent sequences using transformers to capture inter-agent relationships, leveraging masked self-attention to learn relevant agent-level representations. Similarly, TrafficBots [124] uses attention mechanisms to query a shared vectorized context while maintaining individual "personality" encodings, achieving computational efficiency through dot-product

attention and specialized positional encoding, though its domain-specific design limits broader applicability beyond autonomous driving.

Graph-Based Approaches When representing multiple agents in centralized settings, GNN models [135] offer architectural advantages through their inherent ability to model relational structures. Unlike all-to-all approaches that scale quadratically with the number of agents, GNNs can achieve linear scaling in the number of edges in the interaction graph. For example, Structured Multi-Agent World Models (SMAWM) [81] use a GNN to create a factorized state representation where each agent is a node and interactions are edges, achieving better parameter efficiency than all-to-all alternatives. Building on this factorization principle and Iso-Dream's [85] approach to disentanglement, Value Decomposition Framework with Disentangled World Model (VDFD) [115] decomposes its graph-based WM into specialized branches using a Q-Mix-style [91] mixing network and graph convolutions. This modular structure separates controllable dynamics, passive environmental changes, and static features while maintaining the ability to perform multi-step latent rollouts for trajectory imagination. While still centralized in its information sharing, this structured approach demonstrates strong performance on complex tasks with heterogeneous agents.

3.1.2 Decentralized Multi-Agent World Models (D-MAWMs)

D-MAWMs address centralization limitations by decomposing the joint WM into N distributed models, where each agent $i \in \mathcal{I}$ maintains its own model, $\mathcal{M}^i$. Each local model processes observations and maintains a latent state that represents the agent's understanding of the environment. As discussed in the following sections, this decentralization has the potential to reduce action space size and sample complexity at the cost of increased computational and communication complexity. D-MAWMs enable independent decision making in scenarios where centralized control is impractical due to communication constraints or computational limitations [31].

Decentralized Multi-Agent World Models (D-MAWMs) with No Communication In scenarios where direct communication between agents is impossible or impractical, D-MAWMs must employ alternative coordination mechanisms [51, 57, 62, 72, 113, 122, 127]. The literature reveals three primary approaches for achieving coordination without explicit communication channels.

The first approach employs *Opponent Modeling*, where agents predict the behavior of other agents through observation rather than direct communication. For example, Wang et al. [113] demonstrates this in multi-agent rendezvous tasks where agents infer others' intentions through their observed trajectories. Indarjo et al. [51] implement this approach through Deep State Space Models (DSSMs), where each agent maintains predictive models of the actions of other agents using latent variable models and variational inference. The second approach, *stigmergic coordination*, enables agents to react to and influence others implicitly through observable changes in the environment [46]. Xie et al. [122] extend this concept by introducing a RL framework that learns latent representations of opponent strategies, enabling agents to not only predict but also actively influence other agents' policies. Their approach captures high-level latent strategies from low-level actions and adjusts to changes in these strategies over time, enabling the ego agent to influence and guide the opponent's policy effectively. This enables coordination (or manipulation) through anticipation rather than explicit message passing. The third approach leverages *learned emergence* through CTDE, where agents develop implicit coordination protocols during centralized training that transfer to decentralized execution. Kim et al. [57] demonstrates how Masked reconstruction task with QMIX (M-QMIX) learns such protocols, allowing agents to coordinate without runtime communication by sharing information during the training phase. Continuing this concept, Venugopal et al. [111] use model factorization approaches to enable this emergence. Multi-Agent Bi-Level world model (MABL) introduces hierarchical factorization to separate global and agent-specific information:

$$z_t^{g,i} \sim q_\psi(z_t^{g,i} | s_t, z_t^{a,i}, h_t^{g,i}) \qquad \text{(global state)} \tag{4}$$

$$z_t^{a,i} \sim q_\psi(z_t^{a,i} | o_t^i, h_t^{a,i}) \qquad \text{(agent state)} \tag{5}$$

where $z_t^{g,i}$ represents agent i's global state and $z_t^{a,i}$ represents its local state at time t. This structure enables coordination by learning local policies that implicitly encode global information learned during training, thereby enabling enhanced, communication-less emergent behaviors.

While these communication-free approaches eliminate bandwidth and latency constraints, they face several challenges. Firstly, this creates increased computational complexity from maintaining predictive models of other agents [21], which can suffer from non-stationarity issues. When predictive models are not used, this puts the burden on the

training process and implies higher sample complexity during training [22, 77] to learn effective implicit coordination. Lack of communication at inference time can potentially reduce coordination effectiveness in highly dynamic or adversarial scenarios where agent behavior prediction becomes unreliable.

Decentralized Multi-Agent World Models (D-MAWMs) with All-to-All Communication Fully connected communication architectures address the limitations of communication-free approaches by enabling direct information sharing between all agents [23, 58, 69, 98, 112, 120, 124, 125, 131, 133]. Adding explicit communication to the reasoning process allows agents to share their local observations, intentions, and learned models, potentially leading to more coherent and effective collective behavior. In these architectures, each agent can transmit messages to every other agent, enabling the exchange of local observations, intentions, and model predictions. This explicit communication theoretically approaches the performance of centralized systems while maintaining the fault tolerance of decentralized architectures. However, this assumes exponential action scaling and the communication overhead scales quadratically with the number of agents.

Transformer architectures [110] have emerged as a dominant approach for implementing these all-to-all communication schemes [23, 50, 69, 112, 120]. For example, Multi-Agent Model-Based Approach (MAMBA) [23], Models as AGents (MAG) [120], Multi-Agent Temporal Difference MPC (MA-TDMPC) [112], Multi-Agent Policy Optimization with Latent Space Optimization (MAPO-LSO) [50], and Multi-Agent auto-Regressive Imagination for Efficient learning (MARIE) [132] all employ self-attention mechanisms to weight message importance between agents, with each maintaining local WMs that merge agent messages into a coherent world state through transformer-based communication layers[1]. While transformer-based approaches demonstrate effectiveness in coordinating agent behaviors, they introduce two key limitations. First, the transformer communication layers may not map efficiently to real-world communication constraints and protocols. Workarounds can be implemented by performing all-to-all communication and replicating the transformer calculations on each agent, leading to computational inefficiencies. Second, the quadratic compute scaling of attention mechanisms compounds the inherent scaling challenges of all-to-all communication, creating bottlenecks in large-scale deployments (e.g., [19]). These limitations have motivated research into structured communication approaches, such as the graph-based architectures discussed in the following section.

Decentralized Multi-Agent World Models (D-MAWMs) with Graph-Based Communication Graph-based communication architectures generalize previous approaches by representing communication topologies as graphs where vertices represent agents and edges define permitted communication channels [7, 90, 107], thereby subsuming both no-communication and all-to-all communication paradigms. More importantly, it enables representation of realistic communication constraints where agents communicate only with neighbors within communications range or through specific network structures. Other benefits include modeling multi-hop message propagation between non-adjacent agents through intermediate nodes, and time-varying edges can model dynamic network topologies where communication links change based on agent proximity or environmental conditions. Pretorius et al. [90] leverage graph-based communication through differentiable message passing networks for sharing predicted future trajectories generated by their WMs to enhance coordination. A rich messages structure is learned that enables coordination by encoding both current states and predicted futures. Through experiments on digit prediction and invisible navigation tasks, they showed that communicating imagined futures significantly enhanced multi-agent coordination compared to model-free approaches.

While graph-based communication offers significant flexibility, it introduces three key challenges. First, additional algorithmic complexity is required to operate effectively in complex communication topologies [50]. Second, multi-hop communication introduces latency proportional to path length, potentially degrading coordination in time-sensitive scenarios. This also creates conditions where multiple messages must be encoded in each message, creating latent bottlenecks at central nodes [3]. Third, dynamic graph structures require mechanisms to handle topology changes without disrupting ongoing coordination. Addressing these challenges while maintaining the advantages of graph-based communication remains an active area of research.

[1]Note, MAMBA [23] and its derivatives (e.g., MAPO-LSO [50]) can implement graph-based communication through transformer attention mechanisms masked by the adjacency matrix of the communication graph. This masking ensures attention follows the graph structure while leveraging transformer efficiency for message computation. Unfortunately, this masking approach still exhibits $O(n^2)$ compute scaling, and therefore does not take full advantage of the primary benefits of graph-based representations.

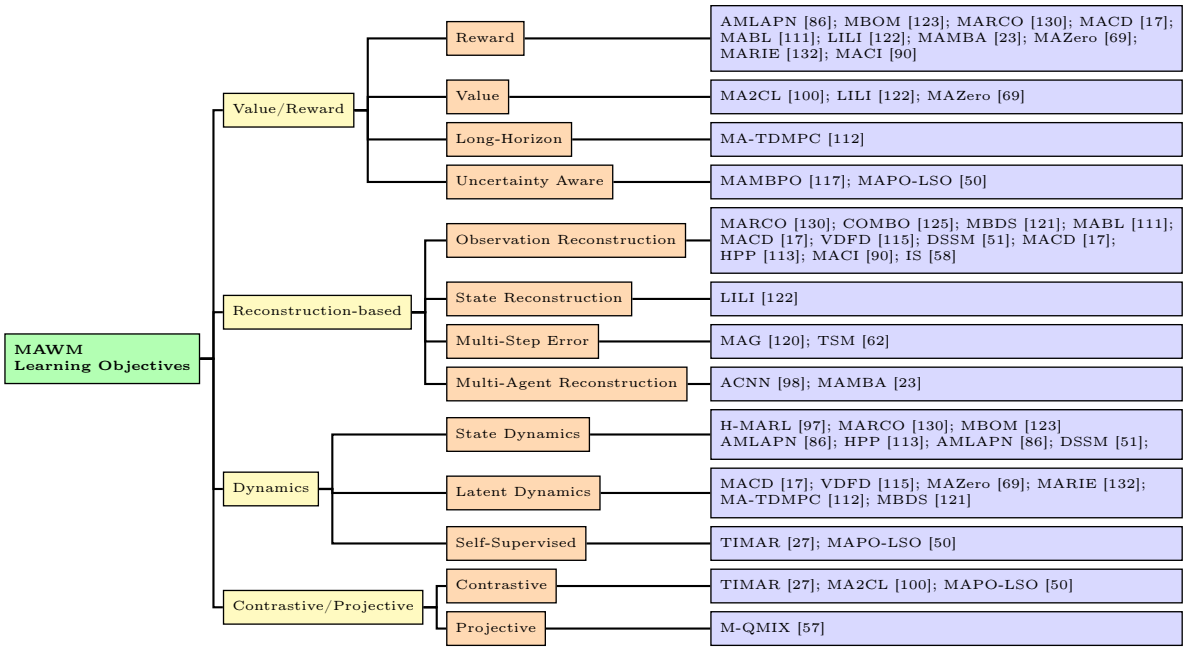

Figure 4: Taxonomy of Learning Objectives in World Models. The hierarchy categorizes approaches by their fundamental learning mechanisms. Reconstruction-based approaches focus on reproducing observations or messages. Value/reward methods emphasize predicting future returns, either directly or through counterfactual reasoning. Latent dynamics approaches model state transitions at different scales (global, local, or factored). Contrastive and projective methods learn representations through comparison or prediction tasks.

3.2 Learning Objectives in Multi-Agent World Models (MAWMs)

MAWMs employ several distinct learning objectives to acquire and maintain world representations, and they play a crucial role in defining the capabilities and performance of MAWMs. For task-specific WMs, reward prediction objectives encourage the model to focus on task-relevant features. Another common objective is direct reconstruction of world states or observations through an auto-encoder framework. While straightforward, this approach can be computationally inefficient by reconstructing irrelevant environmental features. Latent dynamics learning avoids reconstruction by focusing on capturing state transitions in a compressed representation space. Other methods include contrastive approaches, which learn by minimizing the distance between similar states while maximizing it between dissimilar ones, and projective approaches, which predict relationships between similar or corrupted observations in the latent space. In practice, multiple objectives are often combined to achieve better performance. The following section explores these categories of learning objectives used in MAWMs, their principles, advantages, and limitations.

3.2.1 Value and Reward Prediction

Value and reward prediction objectives focus on the model's ability to anticipate future rewards or estimate the value of states and actions [17, 23, 50, 69, 86, 90, 97, 100, 111, 112, 122, 123, 130, 132, 133]. These objectives are particularly useful in RL contexts, where the ultimate goal is to maximize cumulative rewards, and the theoretical results show improved learning rates in reward-aware cases [127]. Generic reward or value prediction losses can be formulated as:

$$\mathbb{L}_{\text{reward}} = \mathbb{E}_{(o_t^i, a_t^i, r_t^i)\sim D}\left[\left(r_t^i - \mathcal{R}_\theta(\mathcal{E}(o_t^i), a_t^i)\right)^2\right] \quad \text{(reward loss)} \tag{6}$$

$$\mathbb{L}_{\text{value}} = \mathbb{E}_{(o_t^i, a_t^i, r_t^i)\sim D}\left[\left(\sum_{k=t}^{t+H-1} \gamma^k r_k^i + \gamma^H V_\theta(\mathcal{E}(o_{t+H}^i)) - V_\theta(\mathcal{E}(o_t^i))\right)^2\right] \quad \text{(value loss)} \tag{7}$$

where o_t^i is agent i's observation at time t, a_t^i is agent i's action, r_t^i is the reward received by agent i, $\hat{r}(\mathcal{E}(o_t^i), a_t^i)$ is the predicted reward from the world model, V_θ is the learned value function, and D is the dataset of experiences.

Value and reward prediction objectives have the advantage of directly aligning the WM with the goal of maximizing

performance in RL tasks. However, they may lead to models that are overly specialized to specific reward structures, potentially limiting generalization to new tasks or environments. For example, Egorov and Shpilman [23] argue that reward-agnostic communication through WMs naturally describes the environment state, while goal-oriented protocols focus on task-specific information exchange. They draw the connection to representation versus acquisition theories in language development [24], where each have their advantages.

Long-Horizon Value Prediction Hansen et al. [43] introduce TD-MPC, which combines a learned terminal value function V_θ with short-horizon planning to estimate returns beyond the horizon H:

$$J^i(\tau^i_{1:H}) = \sum_{t=1}^{H} r^i_t + \gamma^H V_\theta(\mathcal{E}(o^i_H)) \qquad \text{(long-horizon opt.)} \tag{8}$$

In addition to creating reward-informed latent features, this approach allows the WM to make predictions beyond the immediate planning horizon, enhancing long-term decision-making, a limit of MPC approaches. TD-MPC2 [42] expands on this by integrating normalization and architectural innovations that enable stable learning across diverse domains, scaling effectively to a 317M parameter model trained on uncurated multi-domain datasets.

Uncertainty-Aware Reward Modeling Others take an uncertainty-aware approach to reward modeling. By predicting rewards through an ensemble (e.g., Multi-Agent Model-Based Policy Optimization (MAMBPO) [117]) or Monte Carlo dropout (e.g., MAPO-LSO [50]), WMs can estimate uncertainty and generate less biased synthetic data for training, improving sample efficiency. Ablation studies [50] demonstrate that accurate reward modeling is crucial for effective WM learning.

3.2.2 Reconstruction-based Objectives

Reconstruction-based objectives focus on the model's ability to accurately reproduce the observed state of the environment, including the states of all agents [17, 23, 35–40, 51, 58, 68, 85, 90, 98, 105, 111, 113, 115, 116, 121, 122, 125, 128, 129, 133]. This approach is fundamental in ensuring that the WM captures the essential features of the environment and agent interactions. A typical reconstruction loss for a WM can be formalized as:

$$\mathbb{L}_{\text{recon}} = \mathbb{E}_{(o^i_t)\sim D}\left[\|\mathcal{D}(\mathcal{E}(o^i_t)) - o^i_t\|^2_2\right] \qquad \text{(reconstruction loss)} \tag{9}$$

where $\hat{o}^i_t$ is the predicted observation, o^i_t is the true observation, and is the dataset D of observation trajectories sampled from Ω. However, traditional metrics such as mean squared error equally weigh all observation dimensions, forcing models to use capacity on task-irrelevant features [103].

Multi-Agent Reconstruction Objectives Multiple authors extend reconstruction objectives to multi-agent settings through masked-agent attention modules, enabling inter-predictive learning between agents' states [50, 98]. Shang et al. processes unlabeled individual agent observations through a shared NN and learned attention to perform masked observation reconstruction. Similarly, MAPO-LSO employs MA-Self-Predictive Learning (MA-SPL) where masked reconstruction is used to learn agent representations through contrastive learning [50]. Additionally, MAWMs face non-stationarity in the reconstructions as other agents update their behaviors [130].

Multi-Step Reconstruction Error One challenge with learned models is that local prediction errors can propagate over time, leading to large global errors in multi-step rollouts. Krupnik et al. [62] address this by learning patterns across entire sequences of agent behaviors rather than just predicting the next state. Their method uses a Conditional Variational AutoEncoder (CVAE) architecture that first learns to encode observed sequences of agent behaviors into a compact representation. Then, it learns to decode these patterns back into plausible future sequences of agent actions and observations. By training on complete sequences rather than individual transitions, the model better maintains consistency over time and reduces accumulation of small errors that plague single-step approaches. This enables generation of realistic, coherent trajectories that respect both immediate physical constraints and longer-term behavior patterns. Wu et al. [120] take an alternate approach through their MAG framework, which treats local models as decision-making agents and current policies as environment dynamics. Rather than simply minimizing one-step prediction errors, MAG treats the WM as a *policy*, enabling local models to consider multi-step mutual effects by MPC planning the predictions using rollouts. Using random-sampling shooting, each local model

selects predictions that minimize the accumulated errors across the ensemble. The authors prove that minimizing accumulated model errors provides stronger performance guarantees, and they demonstrate the value of treating model learning as a multi-agent optimization problem to minimize accumulated prediction errors.

3.2.3 Dynamics Learning

Dynamics learning objectives focus on capturing the underlying dynamics of the environment and agent interactions in a compact latent space [17, 27, 38–40, 42, 43, 50, 51, 62, 68, 69, 85, 86, 97, 112, 115, 120, 121, 128, 129, 133]. This approach avoids complex reconstructive objectives through the creation of an efficient latent space in which state transitions can be predicted based on their actions and still be reward-/task-agnostic. A typical latent dynamics learning objective can be formulated as:

$$\mathbb{L}_{\text{dynamics}} = \mathbb{E}_{(o_t^i, a_t^i, o_{t+1}^i)\sim D}\left[\|\mathcal{T}_\theta(\mathcal{E}(o_t^i), a_t^i) - \mathcal{E}(o_{t+1}^i)\|_2^2\right] \qquad \text{(dynamics loss)} \tag{10}$$

where $\mathcal{E}$ is the observation encoder and $\mathcal{T}_\theta$ is the learned dynamics function.

Self-Supervised Dynamics Learning This objective encourages the WM to learn latent state evolution without explicit labels on the data. For example, Feng et al. [27] propose Transition-Informed Multi-Agent Representations (TIMAR), which uses a joint transition model to learn latent dynamics through a self-supervised, masked learning approach. The results demonstrate superior performance and data efficiency against MARL benchmarks, and it also improved the robustness and generalization of Transformer-based MARL algorithms such as Multi-Agent Transformer (MAT). Huh and Mohapatra [50] introduce MAPO-LSO, which combines multi-agent transition dynamics reconstruction MA-Transition Dynamics Reconstruction (MA-TDR) and self-predictive learning MA-SPL in a unified learning framework. MA-TDR uses recurrent modeling and predictive representation learning to ground latent states in environment dynamics, while MA-SPL ensures that latent states can predict future states through masked reconstruction, forward dynamics modeling, and inverse dynamics modeling. Unique to MAPO-LSO, Huh and Mohapatra also perform the inverse dynamics learning to predict the action taken given two latent states ℓ_t and ℓ_{t+1} The approach demonstrates significant improvements in sample efficiency (+285.7%) and convergence (+35.68%) across diverse multi-agent tasks when integrated with MARL algorithm baselines.

3.2.4 Contrastive and Projective Learning

Contrastive and projective learning objectives offer powerful approaches to representation learning by capturing meaningful patterns in states or trajectories without relying on full state reconstruction or explicit reward signals [37, 50, 57, 100]. While both contrastive and projective learning aim to learn meaningful representations, they differ in their approach: contrastive learning explicitly maximizes similarity between related samples while minimizing similarity between unrelated ones, whereas projective learning learns representations by predicting missing or corrupted information.

Contrastive Objectives In multi-agent settings, contrastive learning can be used to capture both temporal relationships and inter-agent dependencies by carefully choosing example/counterexamples pairs and the loss function. For example, Song et al. [100] propose Multi-Agent Masked Attentive Contrastive Learning (MA2CL), which uses the following contrastive loss to reconstruct masked agent observations in latent space:

$$\mathbb{L}_{\text{contra}} = \sum_{i\in\mathcal{I}} -\log \frac{\exp(\omega(\mathcal{E}^i(o_t^i), \mathcal{E}^i(o_{t+1}^i)))}{\sum_{j\in\mathcal{I}}\exp(\omega(\mathcal{E}^i(o_t^i), \mathcal{E}^j(o_t^j)))} \qquad \text{(contrastive loss)} \tag{11}$$

where $\omega(\cdot,\cdot)$ is an arbitrary measure of similarity between latent states. This approach enhances agent-level contextual information, improving performance in cooperative tasks. Feng et al. [27] propose TIMAR, which uses a joint transition model to learn effective representations for MARL. The model treats individual observations as a masked sequence of global state contexts and processes them through a transformer-based architecture. TIMAR adapts the Bootstrap Your Own Latent space (BYOL) [32] loss comparing the outputs of the joint transition model with target encoder representations, enabling both temporal and agent-level consistency without requiring pixel-based reconstruction. Limitations of contrastive learning include requiring careful selection of positive and negative pairs [63], and the number of negative samples is exponential in the dimensionality of the problem [66]—both significant problems in complex multi-agent setups.

Projective Objectives Projective learning, on the other hand, aims to make the representations consistent by predicting related representations from one another. These related representations can be from two differing observations, observations of multiple modalities, or corrupted/masked observations. This helps in learning features that remain invariant across different views or partial inputs. For example, M-QMIX [57] uses a predictive loss:

$$\mathbb{L}_{\text{proj}} = \mathbb{E}_{(o_t^i, a_t^i, o_{t+1}^i) \sim D} \left[\| \mathcal{P}(\mathcal{E}_\theta(\tilde{o}_t^i)) - \mathcal{E}_\theta(o_t^i) \|_2^2 \right] \qquad \text{(projective loss)} \tag{12}$$

where $\tilde{o}_t$ represents a corrupted observation (e.g., masked agent observations) and $\mathcal{P}$ is a projection function that attempts to predict the true latent representation. This objective encourages the model to learn a latent space in which states are predictable from various perspectives. M-QMIX [57] addresses sample efficiency in MARL by incorporating a masked reconstruction task into Q-Mix's architecture [91]. Using a BYOL-style [32] approach, M-QMIX applies random feature masking to agent observations and learns to reconstruct full observations, enabling more efficient representation learning. Empirical results on the StarCraft Multi-Agent Challenge (SMAC) benchmark [95] highlighting the benefits of auxiliary self-supervised learning tasks in MARL, demonstrating superior performance in 8 out of 11 test scenarios while using only half the training samples. Limitations of projective learning include representation collapse [66] and sensitivity to how corruption is applied to inputs [126].

3.3 Applications of Multi-Agent World Models (MAWMs)

The learning objectives discussed above enable MAWMs to serve multiple practical purposes in MASs, which are analyzed in this section. The first is the alleviation of costly environmental interactions through the generation of synthetic data through an approximate WM. The next is improving both temporal and agent-based credit assignment calculations. Additionally, WMs can improve learned policy features in two main ways: 1) direct input: using WM features as policy inputs or 2) loss augmentation: adding WM terms to the loss function. This second approach, especially in model-free settings, often functions as a pseudo-WM since it blends modeling with policy learning. In an alternative to learning methods above, MAWMs can also be used to perform online planning or control through the learned model, often using the differentiable nature of the neural representation. Finally, Opponent Modeling WMs can be used to predict the actions and effects of other agents. As with the above, many approaches leverage WMs for multiple different tasks, aiming to maximize the utility of the additional computational effort.

3.3.1 Synthetic Data Generation

Many real-world multi-agent scenarios present significant barriers to data collection, including hardware costs, safety concerns, and time constraints. MAWMs can address this challenge through WM-generated synthetic training data [37, 105]. A WM generates synthetic trajectories by predicting next states and rewards from current observations and actions. While this synthetic data introduces computational cost and model bias [117], it enables rapid exploration of the state space at a fraction of real-world data collection cost. The effectiveness depends heavily on trajectory quality [75], measured through metrics like model prediction error [120] $\epsilon_s = \mathbb{E}_{(s,a)\sim D}[\|\hat{s}_{t+1} - s_{t+1}\|_2^2]$, transition distribution shift [130] $\epsilon_{\mathcal{T}} = D_{\text{KL}}(\mathcal{T}_{\text{real}} \| \mathcal{T}_{\text{synth}})$, or return gap [112] $\epsilon_J = |J(\pi) - J_{\hat{P}}(\pi)|$, where ϵ_s is the error in states, $\epsilon_{\mathcal{T}}$ is the difference between transition probability distributions, and ϵ_J is the difference in task-dependent performance based on the model errors.

Improved Synthetic Data Building on Dreamer V2 [38], MAMBA [23] learns environment dynamics in latent space using attention-based communication [110] to predict the next state based on agents' actions and current states of the model. Starting an initial state sampled from the replay buffer, synthetic data is generated based on current policies of the agents in order to deploy on-policy learning algorithms in the latent space. To ensure independence among local models, MAMBA maximizes the mutual information between an agent's latent state and its previous action ($\mathbb{L}_{\text{MI}} = -\ln p_\theta(a_{t-1}^i | h_t^i, z_t^i)$). Multi-Agent Counterfactual Dreamer (MACD) [17] focuses on generating synthetic trajectories for counterfactual advantage calculations for estimating an agent's contributions. Using the synthetic data, the architecture demonstrates superior training stability and cooperation performance across SMAC [95] and Multi-Agent MuJoCo (MA-MuJuCo) [65] benchmarks compared to both model-free and model-based baselines.

Factorized MAWMs Recognizing that the world contains both controllable and uncontrollable factors, Wang and Meger [115] takes a different approach through VDFD, which decomposes its WM into static (state-conditioned), controllable (action-conditioned), and stochastic branches for synthetic trajectory generation. Using VAEs and value-based training, VDFD generates data without requiring domain knowledge, and shows effectiveness on the standard

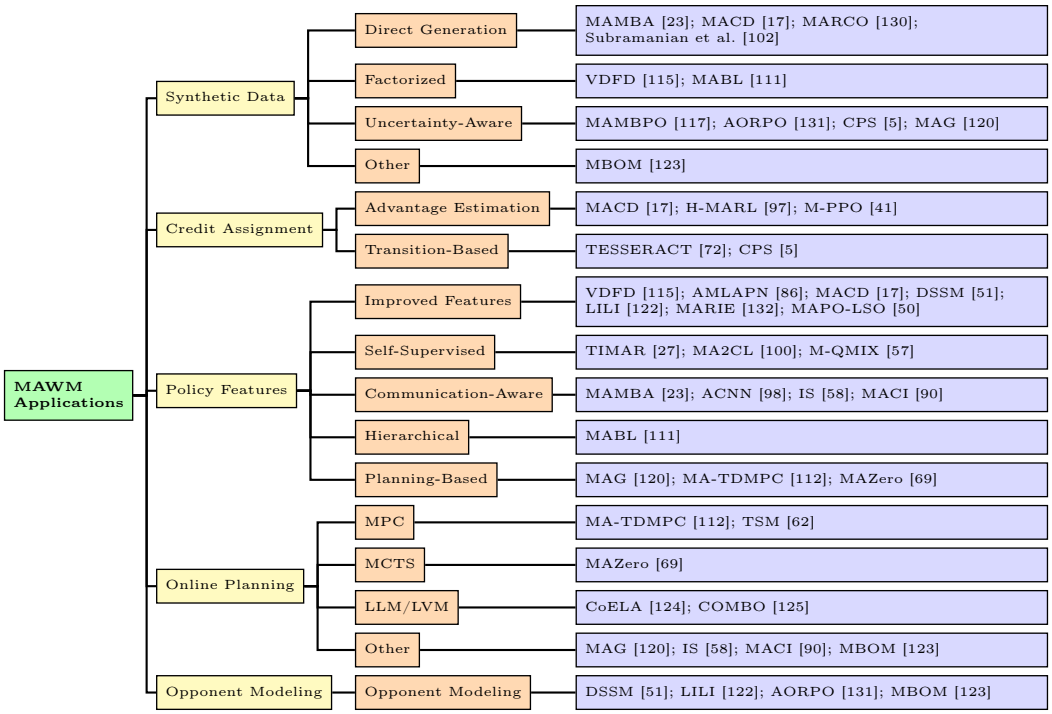

Figure 5: Applications of Multi-Agent World Models. Synthetic data generation methods create additional training examples through model rollouts or counterfactual scenarios. Credit assignment applications leverage world models to better attribute contributions in multi-agent settings. Policy feature extraction approaches learn improved representations for decision-making. Planning and control applications use world models for online decision-making through various methods including MPC, MCTS, and emerging LLM approaches. Opponent modeling applications use world models to predict the actions and effects of other agents.

SMAC benchmarks [95]. Venugopal et al. [111] design MABL specifically for the CTDE setting, where a global WM is used during training for synthetic data generation, and a local WM is used for efficient online action selection. The model generates synthetic trajectories by first computing latent states through the representation model and then using the transition model to predict future states $\hat{z}^{a,i}_{t+1}$ and $\hat{z}^{g,i}_{t+1}$. These trajectories incorporate both global coordination information and local agent dynamics, leading to more effective learning.

Uncertainty-Aware Data Generation Uncertainty-aware modeling is an effective approach to guide synthetic data generation, which—by accounting for the confidence in the transition or reward models—both improves sample efficiency and prevents model bias. Key strategies include driving data collection toward high-variance regions through penalizing uncertain areas, switch to real environments when needed, or actively focusing on states and actions most likely to yield better models [131]. For example, MAMBPO [117] uses an ensemble of stochastic networks to generate synthetic training data, training the network as a maximum likelihood estimator for next observation and rewards. MAMBPO interleaves 10% of real data to avoid overfitting, and this approach achieves significantly 1.7-3.7x better sample efficiency over model-free baselines. Building on this, MARCO [130] employs a dedicated exploration policy shaped by the variance of a learned model. Instead of Dyna-style [105] approaches which relying solely on the current policy for data collection, it targets states and actions where uncertainty is highest, leading to polynomial sample complexity bounds and 12-20x better sample efficiency compared to model-free baselines.

In multi-agent environments, uncertainty can also come from imperfect knowledge of the other agents. Techniques like Adaptive Opponent-wise Rollout Policy Optimization (AORPO) [131] integrate OM with ensemble-based dynamic predictions. By adapting rollout lengths based on model confidence in each opponent's behavior, the method maximizes informative synthetic data generation while curbing model bias, scaling well to complex scenarios. Finally, rather than just quantifying the error, further gains come from multi-step planning to minimize said error. Leveraging Dreamer V2 [38] and MAMBA [23], MAG [120] treats local models as RL decision-makers using the current agent policy as "transition dynamics." By predicting multiple steps ahead, it generates more useful synthetic data by minimizing the compounding error over multiple timesteps.

3.3.2 Credit Assignment

The fundamental challenge of RL is that of credit assignment: identifying and rewarding the actions over time responsible for success. This problem is further complicated in a multi-agent setting because there are now two dimensions to the credit assignment problem—credit assignment over time and credit assignment over agents. WMs enable precise credit assignment in MASs through counterfactual advantage calculations [17, 41, 69] and transition modeling [5, 97].

Advantage Estimation By accurately calculating the advantage, one can understand the benefits of various actions relative to the optimal policy. For example, MACD [17] generates counterfactual trajectories directly through its world model, enabling precise reward allocation and addressing non-stationarity through continuous policy evaluation. The approach estimates agent contributions by comparing expected returns with and without each agent's chosen actions using the WM to simulate counterfactual scenarios. The authors theoretically prove that this counterfactual policy update maximizes the multi-agent learning objective. MACD's advantage estimation demonstrates empirical results on SMAC [95] and MA-MuJuCo [65], due to improved synthetic data quality and credit assignment. In contrast, Han et al. [41] propose Model-Based Credit Assignment (MBCA), a cooperative Multi-Agent PPO (M-PPO) framework that uses WMs to estimate coalition values for credit assignment. By calculating agent i's contribution through a semi-value, it enables more accurate credit assignment by leveraging the WM to evaluate counterfactual contributions across different agent groupings. Empirical results show that agent-specific advantage functions based on semivalues consistently outperform shared global advantage functions in both sample efficiency and final performance. Han et al. demonstrate that the Banzhaf value provides more stable performance than the Shapley value, while both outperform simpler leave-one-out estimation approaches. MAZero [69] enhances credit assignment advantage estimation and the Advantage-Weighted Policy Optimization (AWPO) loss function. This advantage calculation is used by MAZero during MCTS planning to assign credit effectively to branches of the search tree for both exploration and exploitation, particularly in large action spaces where traditional methods struggle.

Transition Knowledge Understanding the transition probabilities can improve training through effective, transition-informed temporal difference calculations. Sessa et al. [97] propose Hallucinated Multi-Agent Reinforcement Learning (H-MARL), a sample-efficient MARL algorithm that constructs high-probability confidence intervals around the unknown transition model. Similar to the Upper Confidence-bound for Trees (UCT) algorithm, H-MARL uses an optimistic hallucinated game to solve the multi-agent equilibria calculations for credit assignment and temporal-difference backpropagation. This approach offers the first guarantees for continuous state and action spaces, ensuring sample-efficient convergence to the equilibria of the underlying Markov game. Bargiacchi et al. [5] propose Cooperative Prioritized Sweeping (CPS) for sample-efficient learning as a generalization the prioritized sweeping (PS) algorithm [78] from single-agent to multi-agent environments. CPS leverages domain knowledge about problem structure through Dynamic Decision Networks (DDNs) to determine which state-action pairs most need updates. The key innovation is using coordination graphs to compute priorities for subsets of joint state-action pairs, avoiding the curse of dimensionality. CPS maintains a priority queue where each pair's priority represents how impactful an update will be to the value function. By factoring priorities across DDN parent sets, CPS can prioritize updates even in environments with hundreds of agents. The authors demonstrate that CPS achieves near optimal performance while outperforming baseline approaches Q-Mix [91] in sample efficiency. Other approaches include Tensorised Actors (TESSERACT) [72], which represents value functions as low-rank tensors (with modes corresponding to different agents' action spaces) to mitigate the exponential growth of the action space. They further extend this approach to a model-based version that uses tensor factorization to estimate the underlying Markov Decision Process (MDP) transitions and rewards, thereby providing an efficient way to improve temporal difference learning and credit assignment.

3.3.3 Policy Feature Extraction

The fundamental challenge in feature-engineered approaches is which features to use. MAWMs avoid this challenge by learning optimal latent features that capture both individual agent dynamics and inter-agent relationships while remaining computationally tractable. Policy feature extraction accomplishes this through two primary methods: 1) using the features online as policy inputs (e.g., $\pi(a_t|o_t) = \pi(a_t|\mathcal{E}(o_t))$) or 2) augmenting loss functions with WM-based terms to improve features by enforcing modeling consistency (e.g., $\mathbb{L}_{\text{policy}} = \mathbb{L}_{\text{task}} + \lambda\mathbb{L}_{\text{model}}$). Both of these approaches tend to blur the line of a traditional world model, and if used in model-free environments, can be categorized as a pseudo-WM (e.g., [86]). This section examines key approaches to policy feature extraction in

MAWMs, including self-supervised objectives, communication-aware representations, planning-based features, and hierarchical representations.

Self-Supervised Features One common approach to learning these latent space representations in MAWMs is through self-supervised masking techniques. These approaches vary in how they apply masking: at the observation level, the agent level, or the global state level. Feng et al. [27] employs global state masking in TIMAR, processing individual observations as masked sequences of the complete state to learn inter-agent relationships through self-attention. These features are then used to generate predictions via joint transition models, demonstrating improved consistency and efficiency in cooperative MARL benchmarks. At the agent level, MA2CL [100] reconstructs masked agent observations using attention and contrastive learning, specifically addressing the challenge of partial, correlated observations in MARL. This enhances the utilization of agent-level contextual information. M-QMIX [57] takes a different approach by incorporating feature-level masking into Q-Mix's architecture [91], allowing agents to learn more robust representations by predicting masked features from their environmental interactions.

Communication-Aware Representations Effective and informative policy features can also be generated through efficient communication protocols. As discussed above, a common approach to this is through the use of attention mechanisms [110] to learn and attend to the important features of messages. Egorov and Shpilman [23] leverage transformers to process the internal states of agents to include critical information from other agents' latent representations. Similarly, Shang et al. [98] investigate the benefits of an object-centric pseudo-WM through an agent-centric attention module with explicit connections across agents and an unsupervised predictive objective to predict future agent states. Another approach combines forward-lookahead planning with attention-based compression. Multiple works develop communicated features from generate imagined trajectories using learned models of environment dynamics and the actions of other agents [58, 90]. In these approaches, agents plan out imagined trajectories using learned models of environment dynamics and the actions of other agents. These trajectories are compressed into messages using attention mechanisms [58] or RNNs [90], allowing agents to share relevant information efficiently.

Hierarchical Representations Hierarchical representations enable multi-scale feature extraction, addressing the challenge of capturing local behavior and global patterns [39]. MABL [111] exemplifies this through its bi-level architecture (see Eqs. (4) and (5)). This separation enables efficient centralized training with synthetic data while maintaining decentralized execution. The learned features benefit from latent information in the global latent space. Empirical results on challenging environments like SMAC [95] demonstrate that these features enable more sample-efficient policy learning compared to centralized and decentralized baselines.

3.3.4 Online Planning and Control

Online planning and control is a key application of MAWMs, where learned models enable agents to predict and optimize their actions in real time while accounting for the behavior of other agents [115, 123]. Notably, leveraging a WM for online planning allows agents to solve novel problems in zero or few-shot scenarios without retraining [134]. However, the exponential growth of joint action spaces with increasing agent count imposes significant computational challenges [49], which are partially mitigated by disentangled [62, 112] and decentralized architectures [98]. Key challenges in multi-agent online planning and control include: (1) partial observability, which complicates state estimation and plan validity [125]; (2) the impact of model accuracy, where prediction errors accumulate over timesteps [62, 77, 120]; and (3) poor scalability of communication and synchronization costs with increasing agent counts [121]. Planning approaches in this domain can be categorized into four primary paradigms: (1) MPC methods that optimize action sequences over finite horizons; (2) MCTS-based techniques that use learned WMs for enhanced rollouts; (3) hybrid approaches combining LLMs with traditional planning frameworks; and (4) other specialized techniques tailored to specific multi-agent scenarios.

MPC Planners Recent work demonstrates the effectiveness of MAWMs with receding horizon MPC planners [58, 62, 69, 112, 113, 120, 121, 124, 125, 133]. For example, MA-TDMPC [112] combines model-based predictions with learned Q-functions to estimate trajectory values during planning (see Eq. (8) above). Unlike conventional MPC methods that optimize locally, MA-TDMPC employs a global communication attention network to coordinate before solving for optimal trajectories in latent space using CEM [93]. Krupnik et al. [62] investigate both cross-gradient

and mutual information factorization techniques for disentangled VAEs to both capture agent interactions and allow for separate optimization of each agent's behavior via Temporal Segment Models (TSMs). This disentangling approach mitigates the error accumulation typical in model-based RL by learning distributions of multi-step trajectory segments, and it simplifies the action space complexity by abstracting away other agents into the latent space. In addition to traditional action optimization, others use MPC to monitor and correct unsafe behaviors. Xiao et al. [121] proposes Model-Based Dynamic Shielding (MBDS), a decentralized MARL framework that uses distributed Linear Temporal Logic (LTL)-based shields to balance scalability and coordination overhead, employing a look-ahead method for real-time synthesis and a WM learning procedure for minimal external knowledge.

MCTS Planners Although MCTS has shown success in world-wide single agent models such as MuZero [96], its application to MAWMs remains limited due to the exponential growth of the joint action space [49]. MAZero [69] adapts MCTS for MAWM planning by using the WM for both state transitions and value estimation during tree search. The approach introduces two key innovations. First, Optimistic Search Lambda (OS(λ)) uses the learned WM to generate trajectories, combining Monte Carlo returns with optimistic value estimates. Second, AWPO guides the tree expansion by weighting actions based on their predicted advantages from the WM. This focuses tree expansion on promising regions of the joint action space, partially mitigating the exponential complexity of multi-agent planning. While MAZero demonstrates the potential of MCTS in MAWMs, combining tree search with WMs in multi-agent settings remains an active area of research.

LLM and Large Vision Model (LVM) Planners With the rise of large generative models, other authors leverage them for online MAWM planning. Cooperative Embodied Language Agent (CoELA) [124] presents a cognitive-inspired modular framework that uses LLMs for perception, memory, communication, and planning. CoELA demonstrated strong performance on cooperative transport tasks through efficient communication and coordination between agents. Similarly, Compositional wOrld Model-based emBOdied (COMBO) [125] introduces a compositional WM that explicitly factors multi-agent dynamics through score-based diffusion models. This compositional structure enables accurate simulation of arbitrary numbers of agents while maintaining computational efficiency. The model employs a two-stage training process with agent-dependent loss scaling to improve multi-agent interaction modeling. To handle partial observability, COMBO uses diffusion models to estimate complete world states from multiple egocentric views.

Other Planners Several approaches leverage planning in novel ways beyond traditional receding-horizon control. MAG [120] embeds MPC within the WM itself, using it to optimize multi-step predictions rather than directly planning actions. This separation of prediction and control planning reduces compounding errors in multi-agent scenarios. Rather than planning directly, Intention Sharing (IS) [58] and Pretorius et al. [90] both propose model-based communication approaches, where agents employ online planning to generate and communicate compressed representations of predicted trajectories using learned WMs. Planning also can be used to enhance offline training processes. Model-Based Opponent Modeling (MBOM) [123] employs offline planning to estimate opponent learning dynamics, improving policy updates through more accurate OM. Similarly, VDFD [115] uses planning-derived features to enhance its Q-Mix-based value function [91], providing more stable training through improved state-value estimation.

3.3.5 Opponent Modeling

Finally, some authors use their MAWMs for other learning tasks, such as Opponent Modeling (OM) [2]. In multi-agent settings, model-based approaches increase in accuracy if the opponents' actions are known, motivating approaches from OM, which attempt to infer opponents' joint policies from observations. Motivated by this, Indarjo et al. [51] propose a formulation of DSSMs in MASs. These models represent environment dynamics from an individual agent's perspective, predicting other agents' actions using latent variable models and variational inference. Xie et al. [122] introduce a RL-based framework for learning latent policy representations, helping the ego agent understand and influence the future strategy of the other agent. It captures high-level latent strategies from low-level actions, adapting over time to guide the opponent's policy towards co-adaptation. Zhang et al. [131] introduce AORPO, a decentralized model-based RL method to address sample complexity by combining dynamics models and opponent models to simulate interactions AORPO uses an ensemble of probabilistic dynamics models and Gaussian opponent policy models. Its key innovation is an adaptive opponent-wise rollout scheme in which the rollout length for each

opponent is computed as based on the policy and the prediction error of opponent j's model. This allows more accurate opponent models to be used for longer rollouts while limiting the influence of less accurate models.

4 Selecting and Implementing MAWMs

Whether to use a MAWM and which type to use is heavily dependent on the challenges of the specific application domain. This section provides a systematic framework for making these decisions, beginning with an analysis of when MAWMs are appropriate and proceeding to detailed architectural choices that affect implementation success.

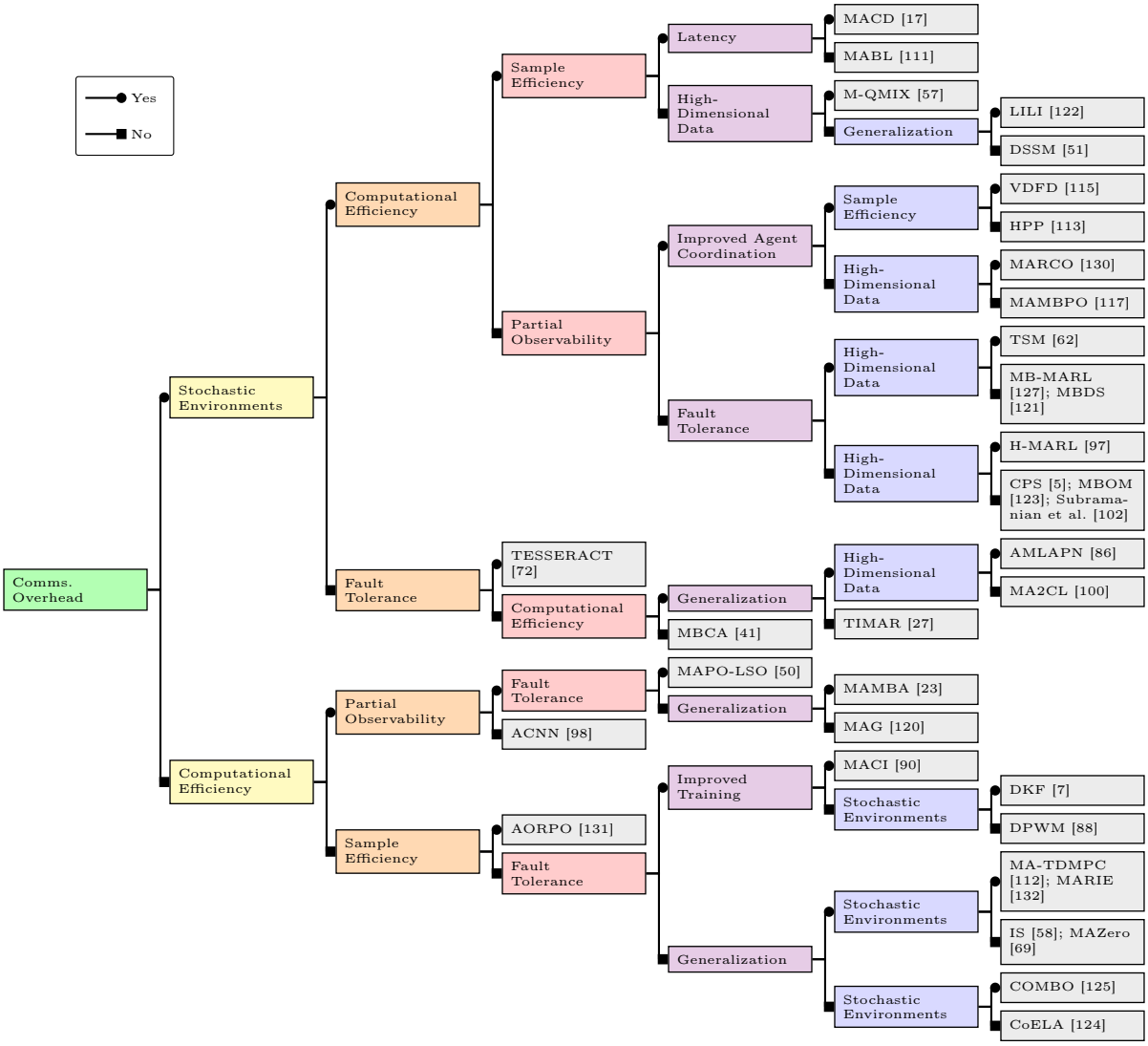

Figure 6: MAWM Selection Framework: A hierarchical taxonomy generated from a dataset of 37 papers using a decision tree classifier. Papers were characterized across 13 capability dimensions and organized to maximize information gain at each split. The resulting tree of depth 6 provides a guide for selecting MAWM implementations based on system requirements, with papers clustered by their demonstrated capabilities. Each node should read as "Is it good for X" with the "Yes" branch going up and the "No" branch going down.

4.1 When to Use MAWMs

Not all applications are suitable for MAWMs. For example, scenarios with fully observable, low-dimensional state spaces and well-understood dynamics are better served by traditional control approaches. For instance, formation control of wheeled robots on flat terrain can be effectively handled by simple potential field methods [9]. As such, the decision to implement a MAWM requires systematic evaluation of the costs and benefits of MAWMs.

4.1.1 MAWM Benefits

MAWMs offer several key advantages over alternative approaches, particularly in scenarios requiring sample efficiency, coordination, or task generalization:

- **Improved Sample Efficiency:** In applications where samples are costly to collect, MAWMs significantly reduce the need for environment interaction during training (e.g, MAMBPO 1.7-3.7x reduction [117] or MARCO polynomial sample complexity [130]).
- **Improved Training:** MAWMs can also improve the use of training samples through improved credit assignment and better utilization of limited sample data.
- **Improved Online Performance:** MAWMs enable better coordination through better policy features which captures essential multi-agent dynamics, or online planning for long-horizon reward optimization that accounts for other agents' potential actions (e.g., MAZero [69]).
- **Task Generalization:** Having an understanding of the dynamics of the world allows traditional RL agents to move beyond reactive policies and to plan out solutions to novel problems, without training (e.g., MA-TDMPC [112]).

4.1.2 MAWM Costs

Implementing MAWMs introduces several types of overhead that must be considered:

- **Computational Requirements:** When compared to model-free baselines, a learned WM increases the online requirements (both compute and memory) required to find solutions.
- **Prediction Accuracy:** Learned models will have accuracy limitations in underexplored areas of the state-action space. This can induce instabilities in the training and/or online performance.
- **Model Validation and Interpretation:** Whereas feature-engineered models have human-interpretable features, this is not the case in learned models, providing validation challenges.

4.1.3 Selection Criteria

If the problem requires improved sample efficiency, training, online performance, or task generalization, then a MAWM may be a good solution, and vice versa. Alternatively, if it is compute constrained or is safety-critical requiring stringent accuracy requirements or formal verification, MAWMs may not be a good solution. In the case that the benefits above do not outweigh the costs, model-free approaches or feature-engineered models may be an ideal solution.

4.2 MAWM Selection

Once the decision to use a MAWM is made, selecting the architecture requires careful consideration of system requirements and constraints. This section provides Fig. 6, a decision tree that guides a practitioner through series of heuristics based on problem attributes to the right class of MAWMs. The decision tree is derived from a systematic analysis of the 37 surveyed approaches across 13 key dimensions: communication overhead, computational efficiency, sample efficiency, observation robustness, generalization capability, fault tolerance, latency requirements, high-dimensional data handling, improved coordination, improved training, stochastic environments, and partial observability. Each approach was manually coded against these dimensions based on demonstrated capabilities in the literature. The tree structure was then optimized using the Gini impurity criterion [89] to maximize information gain at each split, resulting in the presented hierarchy for MAWM selection. Starting with bandwidth limitations/algorithmic communication overhead, each path through the tree represents a different set of key requirements, leading to relevant implementations. For example, in scenarios with strict communication overhead constraints and stochastic environments, but where computational and sample efficiency are priorities, approaches like MACD [17] (when latency is important) or MABL [111] (when latency is less critical) provide suitable solutions. Alternatively, in environments without communication constraints but requiring computational efficiency under partial observability, MAPO-LSO [50] (when fault tolerance is needed) or MAMBA [23] (when fault tolerance is not needed and generalization is the priority) offer effective approaches.

The root node of "Communications Overhead" suggests this is the most important factor in choosing a MAWM approach, directing to either a C-MAWM or a D-MAWM with no communications when communications are constrained and a D-MAWM with either all-to-all or graph-based communications when bandwidth is available. The tree reveals a fundamental tension between observation robustness and computational efficiency. Given a need for communication efficiency, observation robustness is a critical distinguishing characteristic, with roughly equal papers focusing on robust approaches (e.g., value/reward [86, 97, 130] or contrastive/projective approaches [27, 100]) and those focusing on non-robust approaches (e.g., reconstruction [17, 51, 111, 113, 121, 122]). Alternatively, papers that do not require efficient communications overhead are better divided by their requirement for computational efficiency: MAMBA [23] and MAPO-LSO [50] provide good computational efficiency, whereas approaches like MA-TDMPC [112] and MAZero [69] eschew efficiency for other parameters line improved online performance.

The depth of the tree (6) and the distribution of papers across leaves demonstrates that no approach excels across all dimensions, suggesting practitioners must carefully weigh these competing concerns based on the specific use case. Most papers address specific combinations of requirements (e.g., "sample efficiency + stochastic environments + generalization") rather than providing universal solutions, indicating significant future work as discussed in the following section.

5 Future Directions

The analysis of existing MAWMs reveals that—despite significant advances in architectures and learning approaches—key limitations persist. Addressing these challenges defines critical research directions for advancing multi-agent world modeling capabilities.

Model Scalability and Computational Complexity MAWMs face fundamental scalability challenges in both model size and agent count, with computational requirements scaling as $O(|\mathcal{A}|^N)$ in centralized architectures [45]. While approaches like tensor factorization [72] and approximate models [130] reduce this complexity, they introduce accuracy trade-offs that compound over time [120]. Graph-based architectures [50, 81] offer promise through explicit connectivity modeling, though distributed consistency remains challenging. Recent work establishes power-law scaling relationships between model size, data, and performance [42, 87], but these findings are limited to single-agent scenarios. Hierarchical approaches across state [111], agent [133], and temporal [39] dimensions show potential for balancing fine-grained interactions with high-level dynamics, though integrating these abstractions effectively remains an open challenge.

Task-Agnostic vs Task-Specific Modeling While WMs enable zero-shot generalization to new tasks, many MAWMs remain task-specific through their focus on reward prediction [117]. This limits their flexibility across objectives and environments. Recent work like M-QMIX [57] demonstrates the potential of task-agnostic approaches that separate environment dynamics from reward structures. However, this separation introduces objective mismatch challenges [64], where improved prediction accuracy may not translate to better task performance. Additionally, MAWMs face challenges in generalizing across differing agent conditions, particularly those involving varying numbers of agents or diverse interactions [41], though approaches using dynamic architectures (e.g., GNNs [50, 81]) show promise. Future work must balance trade-offs between task-specific performance and general applicability while maintaining robustness to changing conditions.

Distributed Modeling Challenges D-MAWMs face challenges in maintaining consistency, especially in partially observable environments with limited communication [90]. Ensuring that individual agents' WMs remain synchronized and coherent with the true environment state—while accounting for the actions and observations of other agents—is a significant challenge. Existing approaches rely heavily on transformers [110] to maintain consistency across latent states [23], which may be difficult to adapt to distributed and real-world applications. Future research should explore efficient and realistic methods for model synchronization and consistency maintenance in decentralized settings, particularly in models that exchange latent representations.

Formal Verification and Interpretability One of the benefits of using feature-engineered WMs is the ability to apply formal methods for performance and safety guarantees [4, 6, 25]. For example, Xiao et al. [121] demonstrate how LTL specifications can be used to create dynamic shields that monitor and correct unsafe behaviors in MAWMs.

However, extending formal guarantees to learned MAWMs—particularly in safety-critical multi-agent scenarios like autonomous driving and multi-robot systems—remains an open challenge due to the black-box nature of NNs, given that verification of a generic NN is NP-Complete [55]. Additionally, the latent nature of representations makes it challenging to interpret the learned dynamics against human domain knowledge, requiring re-encoding of the latent space into human-readable observations to validate WM predictions [39].

Generative Models in MAWMs The integration of LLMs with WMs opens new possibilities for flexible MASs. CoELA [124] demonstrates improved human-AI cooperation through natural language communication, while COMBO [125] uses LLMs for coordinating multiple agents through compositional WMs. These approaches suggest language models could enhance multi-agent coordination by providing natural interfaces for planning and task specification. In addition, recent advances in generative modeling—such as Sora for video generation [14], Genie for interactive environments [15], or Neural Radiance Fields (NeRF) for observation reconstructions [76]—indicate potential paths for enhancing MAWMs with more sophisticated visual and interactive capabilities.

Real-World Applications Transitioning MAWMs from simulation to physical systems introduces significant challenges, as demonstrated by recent robotics applications. DayDreamer [119] achieved impressive sample efficiency—learning quadruped locomotion in 1 hour without simulation—but highlighted difficulties with sensor noise and environmental variability. Similarly, autonomous driving applications like GAIA-1 [48] emphasize the safety-critical nature of prediction errors in dynamic multi-agent environments. Beyond technical challenges, human interaction introduces additional complexity, though LLM approaches like CoELA [124] show promise in improving trust and task completion through natural communication. Future work must address robust learning from noisy data, partial observability in physical systems, formal safety guarantees, and scalable deployment while maintaining practical computational constraints.

6 Contributions

This survey provides a comprehensive framework for categorizing and comparing MAWMs based on their architectures, learning objectives, and applications, highlighting trade-offs between computational scalability, communication overhead, and coordination effectiveness. Guidelines are provided for selecting communication architectures, tailoring learning objectives to task specificity, and selecting applications for MAWMs. It identifies critical challenges like exponential complexity in centralized architectures, consistency maintenance in decentralized settings with partial observability, and sample efficiency limitations. The survey outlines research directions, such as hierarchical architectures for complexity management, self-supervised learning for sample efficiency, and techniques for distributed consistency, offering both theoretical foundations and insights for advancing scalable and robust MAWMs in real-world systems.

References

[1] Constructions Aeronautiques et al. "Pddl| the planning domain definition language". In: (1998) (cit. on p. 5).

[2] Stefano V Albrecht and Peter Stone. "Autonomous agents modelling other agents: A comprehensive survey and open problems". In: *Artif. Intell.* 258 (May 2018), pp. 66–95. ISSN: 0004-3702. DOI: 10.1016/j.artint.2018.01.002. URL: https://www.sciencedirect.com/science/article/pii/S0004370218300249 (cit. on p. 20).

[3] Uri Alon and Eran Yahav. "On the bottleneck of graph neural networks and its practical implications". In: *arXiv [cs.LG]* (June 2020). arXiv: 2006.05205 [cs.LG]. URL: http://arxiv.org/abs/2006.05205 (cit. on p. 12).

[4] Mohammed Alshiekh et al. "Safe Reinforcement Learning via Shielding". In: *arXiv [cs.LO]* (Aug. 2017). arXiv: 1708.08611 [cs.LO] (cit. on p. 23).

[5] Eugenio Bargiacchi, T Verstraeten, and D Roijers. "Cooperative Prioritized Sweeping". In: *Adapt Agent Multi-agent Syst* (2021), pp. 160–168. DOI: 10.5555/3463952.3463977. URL: https://cris.vub.be/ws/portalfiles/portal/75769165/p160.pdf (cit. on pp. 10, 17, 18, 21).

[6] Osbert Bastani. "Safe Reinforcement Learning with Nonlinear Dynamics via Model Predictive Shielding". In: *arXiv [cs.LG]* (May 2019). arXiv: 1905.10691 [cs.LG]. URL: http://arxiv.org/abs/1905.10691 (cit. on p. 23).

[7] Giorgio Battistelli and Luigi Chisci. "Stability of consensus extended Kalman filter for distributed state estimation". In: *Automatica* 68 (June 2016), pp. 169–178. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2016.01.071. URL: https://www.sciencedirect.com/science/article/pii/S0005109816300188 (cit. on pp. 4, 5, 12, 21).

[8] Leonard E Baum and Ted Petrie. "Statistical Inference for Probabilistic Functions of Finite State Markov Chains". en. In: *aoms* 37.6 (Dec. 1966), pp. 1554–1563. ISSN: 0003-4851,2168-8990. DOI: 10.1214/aoms/1177699147. URL: https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-37/issue-6/Statistical-Inference-for-Probabilistic-Functions-of-Finite-State-Markov-Chains/10.1214/aoms/1177699147.full (cit. on p. 4).

[9] J L Baxter et al. "Multi-robot search and rescue: A potential field based approach". In: *Autonomous Robots and Agents*. Studies in computational intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 9–16. ISBN: 9783540734239,9783540734246 (cit. on p. 21).

[10] Daniel S Bernstein et al. "The complexity of decentralized control of Markov decision processes". In: *Math. Oper. Res.* 27.4 (2002), pp. 819–840. ISSN: 0364-765X,1526-5471. URL: https://www.jstor.org/stable/3690469 (cit. on p. 10).

[11] Daan Bloembergen et al. "Evolutionary Dynamics of Multi-Agent Learning: A Survey". en. In: *jair* 53 (Aug. 2015), pp. 659–697. ISSN: 1076-9757,1076-9757. DOI: 10.1613/jair.4818. URL: https://www.jair.org/index.php/jair/article/view/10952 (cit. on p. 8).

[12] Karim Bouyarmane et al. "Quadratic programming for multirobot and task-space force control". In: *IEEE Trans. Robot.* 35.1 (Feb. 2019), pp. 64–77. ISSN: 1552-3098,1941-0468 (cit. on p. 1).

[13] R Brafman and Carmel Domshlak. "From one to many: Planning for loosely coupled multi-agent systems". In: *Int Conf Autom Plan Sched* (Sept. 2008), pp. 28–35 (cit. on p. 6).

[14] Tim Brooks et al. "Video generation models as world simulators". In: (2024). URL: https://openai.com/research/video-generation-models-as-world-simulators (cit. on pp. 4, 8, 24).

[15] Jake Bruce et al. "Genie: Generative Interactive Environments". In: *arXiv [cs.LG]* (Feb. 2024). arXiv: 2402.15391 [cs.LG]. URL: http://arxiv.org/abs/2402.15391 (cit. on pp. 4, 8, 24).

[16] Cesar Cadena et al. "Past, present, and future of simultaneous Localization and mapping: Towards the robust-perception age". In: *arXiv [cs.RO]* (June 2016). arXiv: 1606.05830 [cs.RO]. URL: http://arxiv.org/abs/1606.05830 (cit. on p. 4).

[17] Jiajun Chai et al. "Aligning credit for multi-agent cooperation via model-based counterfactual imagination". In: *Adapt Agent Multi-agent Syst* (2024), pp. 281–289 (cit. on pp. 9, 10, 13–18, 21–23).

[18] Bradley J Clement and Edmund H Durfee. "Top-down search for coordinating the hierarchical plans of multiple agents". In: *Proceedings of the third annual conference on Autonomous Agents*. New York, NY, USA: ACM, Apr. 1999. ISBN: 9781581130669. DOI: 10.1145/301136.301205. URL: http://dx.doi.org/10.1145/301136.301205 (cit. on p. 6).

[19] Tri Dao et al. "FlashAttention: Fast and memory-efficient exact attention with IO-awareness". In: *Neural Inf Process Syst* abs/2205.14135 (May 2022). Ed. by S Koyejo et al., pp. 16344–16359 (cit. on p. 12).

[20] M Deisenroth and C E Rasmussen. "PILCO: A model-based and data-efficient approach to policy search". In: *of the 28th International Conference on ...* (2011) (cit. on p. 5).

[21] Adam Eck et al. "Scalable decision-theoretic planning in open and typed multiagent systems". In: *arXiv [cs.MA]* (Nov. 2019). arXiv: 1911.08642 [cs.MA]. URL: http://arxiv.org/abs/1911.08642 (cit. on p. 11).

[22] Yonathan Efroni, Mohammad Ghavamzadeh, and Shie Mannor. "Online planning with lookahead policies". In: *arXiv [cs.LG]* (Sept. 2019). arXiv: 1909.04236 [cs.LG]. URL: http://arxiv.org/abs/1909.04236 (cit. on p. 12).

[23] Vladimir Egorov and Alexei Shpilman. "Scalable Multi-Agent Model-Based Reinforcement Learning". In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '22. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, May 2022, pp. 381–390. ISBN: 9781450392136 (cit. on pp. 4, 10, 12–14, 16, 17, 19, 21–23).

[24] N C Ellis and D Larsen-Freeman. "Language emergence: Implications for applied linguistics–introduction to the special issue". In: *Appl. Linguist.* 27.4 (Dec. 2006), pp. 558–589. ISSN: 0142-6001,1477-450X. DOI: 10.1093/applin/aml028. URL: https://academic.oup.com/applij/article/27/4/558/155704 (cit. on p. 14).

[25] Ingy ElSayed-Aly et al. "Safe Multi-Agent Reinforcement Learning via Shielding". In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '21. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, May 2021, pp. 483–491. ISBN: 9781450383073. URL: https://dl.acm.org/doi/10.5555/3463952.3464013 (cit. on p. 23).

[26] K Erol, J Hendler, and D Nau. "HTN Planning: Complexity and Expressivity". In: *AAAI* (1994). URL: https://www.semanticscholar.org/paper/05582e59616c2e5a3aa56e949f6f3feaf95fd316 (cit. on pp. 4, 5).

[27] Mingxiao Feng et al. "Timar: Transition-Informed Representation for Sample-Efficient Multi-Agent Reinforcement Learning". In: *Available at SSRN 4706110* (Dec. 2023) (cit. on pp. 10, 13, 15, 17, 19, 21, 23).

[28] Richard E Fikes and Nils J Nilsson. "Strips: A new approach to the application of theorem proving to problem solving". In: *Artif. Intell.* 2.3 (Dec. 1971), pp. 189–208. ISSN: 0004-3702. DOI: 10.1016/0004-3702(71)90010-5. URL: https://www.sciencedirect.com/science/article/pii/0004370271900105 (cit. on pp. 4, 5).

[29] Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. "Distributed Constraint Optimization Problems and Applications: A Survey". en. In: *jair* 61 (Mar. 2018), pp. 623–698. ISSN: 1076-9757,1076-9757. DOI: 10.1613/jair.5565. URL: https://www.jair.org/index.php/jair/article/view/11185 (cit. on p. 4).

[30] Benyamin Ghojogh and Ali Ghodsi. "Recurrent neural networks and Long Short-term memory networks: Tutorial and survey". In: *arXiv [cs.LG]* (Apr. 2023). arXiv: 2304.11461 [cs.LG]. URL: http://arxiv.org/abs/2304.11461 (cit. on p. 10).

[31] Claudia V Goldman and Shlomo Zilberstein. "Decentralized control of cooperative systems: categorization and complexity analysis". In: *J. Artif. Intell. Res.* 22.1 (Nov. 2004), pp. 143–174. ISSN: 1076-9757. URL: https://dl.acm.org/doi/10.5555/1622487.1622493 (cit. on p. 11).

[32] Jean-Bastien Grill et al. "Bootstrap your own latent: A new approach to self-supervised Learning". In: *arXiv [cs.LG]* (June 2020). arXiv: 2006.07733 [cs.LG]. URL: http://arxiv.org/abs/2006.07733 (cit. on pp. 15, 16).

[33] Sven Gronauer and Klaus Diepold. "Multi-agent deep reinforcement learning: a survey". en. In: *Artif. Intell. Rev.* 55.2 (Feb. 2022), pp. 895–943. ISSN: 0269-2821,1573-7462. DOI: 10.1007/s10462-021-09996-w. URL: https://link.springer.com/article/10.1007/s10462-021-09996-w (cit. on p. 33).

[34] Yanchen Guan et al. "World Models for Autonomous Driving: An Initial Survey". In: *arXiv [cs.LG]* (Mar. 2024). arXiv: 2403.02622 [cs.LG]. URL: http://arxiv.org/abs/2403.02622 (cit. on p. 33).

[35] David Ha and Jürgen Schmidhuber. "World Models". In: *arXiv [cs.LG]* (Mar. 2018). arXiv: 1803.10122 [cs.LG] (cit. on pp. 2–4, 14).

[36] Danijar Hafner et al. "Learning Latent Dynamics for Planning from Pixels". In: *arXiv [cs.LG]* (Nov. 2018). arXiv: 1811.04551 [cs.LG] (cit. on pp. 7, 14).

[37] Danijar Hafner et al. "Dream to Control: Learning Behaviors by Latent Imagination". In: *arXiv [cs.LG]* (Dec. 2019). arXiv: 1912.01603 [cs.LG]. URL: http://arxiv.org/abs/1912.01603 (cit. on pp. 2–4, 7, 14–16).

[38] Danijar Hafner et al. "Mastering Atari with Discrete World Models". In: *arXiv [cs.LG]* (Oct. 2020). arXiv: 2010.02193 [cs.LG]. URL: http://arxiv.org/abs/2010.02193 (cit. on pp. 2, 4, 7, 14–17).

[39] Danijar Hafner et al. *Deep Hierarchical Planning from Pixels*. June 2022. eprint: 2206.04114v1. URL: http://arxiv.org/abs/2206.04114v1 (cit. on pp. 7, 14, 15, 19, 23, 24).

[40] Danijar Hafner et al. "Mastering Diverse Domains through World Models". In: *arXiv [cs.AI]* (Jan. 2023). arXiv: 2301.04104 [cs.AI]. URL: http://arxiv.org/abs/2301.04104 (cit. on pp. 4, 7, 14, 15).

[41] Dongge Han et al. "Multiagent Model-based Credit Assignment for Continuous Control". In: *arXiv [cs.AI]* (Dec. 2021). arXiv: 2112.13937 [cs.AI]. URL: http://arxiv.org/abs/2112.13937 (cit. on pp. 10, 17, 18, 21, 23).

[42] Nicklas Hansen, Hao Su, and Xiaolong Wang. "TD-MPC2: Scalable, Robust World Models for Continuous Control". In: *arXiv [cs.LG]* (Oct. 2023). arXiv: 2310.16828 [cs.LG]. URL: http://arxiv.org/abs/2310.16828 (cit. on pp. 8, 14, 15, 23).

[43] Nicklas Hansen, Xiaolong Wang, and Hao Su. "Temporal Difference Learning for Model Predictive Control". In: *arXiv [cs.LG]* (Mar. 2022). arXiv: 2203.04955 [cs.LG]. URL: http://arxiv.org/abs/2203.04955 (cit. on pp. 4, 8, 14, 15).

[44] Matthew Hausknecht and Peter Stone. "Deep Recurrent Q-Learning for Partially Observable MDPs". In: *arXiv [cs.LG]* (July 2015). arXiv: 1507.06527 [cs.LG]. URL: http://arxiv.org/abs/1507.06527 (cit. on p. 3).

[45] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. "A survey and critique of multiagent deep reinforcement learning". In: *Auton. Agent. Multi. Agent. Syst.* 33.6 (Nov. 2019), pp. 750–797. ISSN: 1387-2532,1573-7454. DOI: 10.1007/s10458-019-09421-1. URL: https://doi.org/10.1007/s10458-019-09421-1 (cit. on pp. 2, 10, 23).

[46] Francis Heylighen. "Stigmergy as a universal coordination mechanism I: Definition and components". en. In: *Cogn. Syst. Res.* 38 (June 2016), pp. 4–13. ISSN: 2214-4366,1389-0417. DOI: 10.1016/j.cogsys.2015.12.002. URL: https://www.sciencedirect.com/science/article/pii/S1389041715000327 (cit. on p. 11).

[47] Anthony Hu et al. "GAIA-1: A Generative World Model for Autonomous Driving". In: *arXiv [cs.CV]* (Sept. 2023). arXiv: 2309.17080 [cs.CV]. URL: http://arxiv.org/abs/2309.17080 (cit. on p. 1).

[48] Anthony Hu et al. "GAIA-1: A generative world model for autonomous driving". In: *arXiv [cs.CV]* (Sept. 2023). arXiv: 2309.17080 [cs.CV]. URL: http://arxiv.org/abs/2309.17080 (cit. on p. 24).

[49] Thomas Hubert et al. "Learning and Planning in Complex Action Spaces". In: *arXiv [cs.LG]* (Apr. 2021). arXiv: 2104.06303 [cs.LG] (cit. on pp. 19, 20).

[50] Dom Huh and Prasant Mohapatra. "Representation Learning For Efficient Deep Multi-Agent Reinforcement Learning". In: *arXiv [cs.MA]* (June 2024). arXiv: 2406.02890 [cs.MA]. URL: http://arxiv.org/abs/2406.02890 (cit. on pp. 10, 12–15, 17, 21–23).

[51] Pararawendy Indarjo, Michael Kaisers, and P D Grunwald. "Deep state-space models in multi-agent systems". In: *Master's thesis, Leiden University* (2019) (cit. on pp. 10, 11, 13–15, 17, 20, 21, 23).

[52] L Kaelbling and Tomas Lozano-Perez. "Integrated task and motion planning in belief space". In: *The International Journal of Robotics Research* 32 (Aug. 2013), pp. 1194–1227. DOI: 10.1177/0278364913484072. URL: https://lis.csail.mit.edu/pubs/tlp/IJRRBelFinal.pdf (cit. on p. 5).

[53] Thomas Kailath. *Linear Systems*. en. Prentice-Hall information and system sciences series. Upper Saddle River, NJ: Pearson, Nov. 1979. ISBN: 9780135369616 (cit. on p. 5).

[54] R E Kalman. "A New Approach to Linear Filtering and Prediction Problems". en. In: *J. Basic Eng* 82.1 (Mar. 1960), pp. 35–45. ISSN: 0021-9223. DOI: 10.1115/1.3662552. URL: https://asmedigitalcollection.asme.org/fluidsengineering/article/82/1/35/397706/A-New-Approach-to-Linear-Filtering-and-Prediction (cit. on p. 4).

[55] Guy Katz et al. "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks". In: *Computer Aided Verification*. Springer International Publishing, 2017, pp. 97–117 (cit. on p. 24).

[56] Masoud Khodarahmi and Vafa Maihami. "A review on Kalman filter models". en. In: *Arch. Comput. Methods Eng.* 30.1 (Jan. 2023), pp. 727–747. ISSN: 1134-3060,1886-1784. DOI: 10.1007/s11831-022-09815-7. URL: https://link.springer.com/article/10.1007/s11831-022-09815-7 (cit. on p. 4).

[57] Jung In Kim et al. "Sample-efficient multi-agent reinforcement learning with masked reconstruction". en. In: *PLoS One* 18.9 (Sept. 2023), e0291545. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0291545. URL: http://dx.doi.org/10.1371/journal.pone.0291545 (cit. on pp. 10, 11, 13, 15–17, 19, 21, 23).

[58] Woojun Kim, Jongeui Park, and Youngchul Sung. "Communication in Multi-Agent Reinforcement Learning: Intention Sharing". In: (Oct. 2020). URL: https://openreview.net/pdf?id=qpsl2dR9twy (cit. on pp. 10, 12–14, 17, 19–21).

[59] Thomas Kipf, Elise van der Pol, and Max Welling. "Contrastive learning of Structured World Models". In: *arXiv [stat.ML]* (Nov. 2019). DOI: 10.48550/ARXIV.1911.12247. arXiv: 1911.12247 [stat.ML]. URL: http://arxiv.org/abs/1911.12247 (cit. on p. 7).

[60] Dan Kondratyuk et al. "VideoPoet: A large language model for zero-shot video generation". In: *arXiv [cs.CV]* (Dec. 2023). arXiv: 2312.14125 [cs.CV]. URL: http://arxiv.org/abs/2312.14125 (cit. on p. 8).

[61] B O Koopman. "Hamiltonian systems and transformations in Hilbert space". In: *Proc. Natl. Acad. Sci. U. S. A.* 17.5 (1931), pp. 315–318. ISSN: 0027-8424,1091-6490 (cit. on p. 6).

[62] Orr Krupnik, Igor Mordatch, and Aviv Tamar. "Multi-Agent Reinforcement Learning with Multi-Step Generative Models". In: *Proceedings of the Conference on Robot Learning*. Ed. by Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura. Vol. 100. Proceedings of Machine Learning Research. PMLR, 2020, pp. 776–790 (cit. on pp. 10, 11, 13–15, 17, 19, 21).

[63] Pranjal Kumar, Piyush Rawat, and Siddhartha Chauhan. "Contrastive self-supervised learning: review, progress, challenges and future research directions". en. In: *Int. J. Multimed. Inf. Retr.* 11.4 (Dec. 2022), pp. 461–488. ISSN: 2192-6611,2192-662X (cit. on p. 15).

[64] Nathan Lambert et al. "Objective Mismatch in Model-based Reinforcement Learning". In: *arXiv [cs.LG]* (Feb. 2020). arXiv: 2002.04523 [cs.LG]. URL: http://arxiv.org/abs/2002.04523 (cit. on p. 23).

[65] Rodrigo de Lazcano et al. *Gymnasium Robotics*. Version 1.3.1. 2024. URL: http://github.com/Farama-Foundation/Gymnasium-Robotics (cit. on pp. 16, 18).

[66] Yann LeCun. "A path towards autonomous machine intelligence version 0.9.2, 2022-06-27". In: (2022) (cit. on pp. 4, 8, 15, 16).

[67] Zs Lendek, R Babuška, and B De Schutter. "Distributed Kalman filtering for multiagent systems". In: *2007 European Control Conference (ECC)*. IEEE, July 2007, pp. 2193–2200. ISBN: 9783952417386. DOI: 10.23919/ECC.2007.7068267. URL: https://ieeexplore.ieee.org/document/7068267 (cit. on p. 5).

[68] Yunzhu Li et al. "Learning Compositional Koopman Operators for Model-Based Control". In: *arXiv [cs.LG]* (Oct. 2019). arXiv: 1910.08264 [cs.LG]. URL: http://arxiv.org/abs/1910.08264 (cit. on pp. 6, 14, 15).

[69] Qihan Liu et al. "Efficient Multi-agent Reinforcement Learning by Planning". In: *arXiv [cs.LG]* (May 2024). arXiv: 2405.11778 [cs.LG]. URL: http://arxiv.org/abs/2405.11778 (cit. on pp. 4, 10, 12, 13, 15, 17–23).

[70] Lennart Ljung. "System Identification". In: *Signal Analysis and Prediction*. Ed. by Ales Procházka et al. Boston, MA: Birkhäuser Boston, 1998, pp. 163–173. ISBN: 9781461217688. DOI: 10.1007/978-1-4612-1768-8_11. URL: https://doi.org/10.1007/978-1-4612-1768-8_11 (cit. on pp. 4, 5).

[71] F M Luo et al. "A survey on model-based reinforcement learning". In: *Sci. China Inf. Sci.* (2024). ISSN: 1674-733X. URL: https://link.springer.com/article/10.1007/s11432-022-3696-5 (cit. on p. 33).

[72] Anuj Mahajan et al. "Tesseract: Tensorised Actors for Multi-Agent Reinforcement Learning". In: *arXiv [cs.LG]* (May 2021). arXiv: 2106.00136 [cs.LG]. URL: http://arxiv.org/abs/2106.00136 (cit. on pp. 10, 11, 17, 18, 21, 23).

[73] A Mauroy, Y Susuki, and I Mezić. *The Koopman Operator in Systems and Control*. Springer International Publishing, 2020. DOI: 10.1007/978-3-030-35713-9. URL: https://link.springer.com/book/10.1007/978-3-030-35713-9 (cit. on p. 6).

[74] Igor Mezić. "Spectral Properties of Dynamical Systems, Model Reduction and Decompositions". In: *Nonlinear Dyn.* 41.1 (Aug. 2005), pp. 309–325. ISSN: 0924-090X,1573-269X. DOI: 10.1007/s11071-005-2824-x. URL: https://doi.org/10.1007/s11071-005-2824-x (cit. on p. 6).

[75] Vincent Micheli, Eloi Alonso, and François Fleuret. "Transformers are Sample-Efficient World Models". In: *arXiv [cs.LG]* (Sept. 2022). arXiv: 2209.00588 [cs.LG]. URL: http://arxiv.org/abs/2209.00588 (cit. on pp. 8, 16).

[76] Ben Mildenhall et al. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *Computer Vision – ECCV 2020*. Springer International Publishing, 2020, pp. 405–421. DOI: 10.1007/978-3-030-58452-8_24. URL: http://dx.doi.org/10.1007/978-3-030-58452-8_24 (cit. on p. 24).

[77] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. "Model-based Reinforcement Learning: A Survey". In: *arXiv [cs.LG]* (June 2020). arXiv: 2006.16712 [cs.LG]. URL: http://arxiv.org/abs/2006.16712 (cit. on pp. 8, 12, 19, 33).

[78] Andrew W Moore and Christopher G Atkeson. "Prioritized sweeping: Reinforcement learning with less data and less time". en. In: *Mach. Learn.* 13.1 (Oct. 1993), pp. 103–130. ISSN: 0885-6125,1573-0565. DOI: 10.1007/bf00993104. URL: https://link.springer.com/article/10.1007/BF00993104 (cit. on p. 18).

[79] Kumpati S Narendra and Anuradha M Annaswamy. *Stable Adaptive Systems*. en. Courier Corporation, July 2012. ISBN: 9780486141428 (cit. on pp. 4, 5).

[80] D S Nau et al. "SHOP2: An HTN Planning System". en. In: *J. Artif. Intell. Res.* 20 (Dec. 2003), pp. 379–404. ISSN: 1076-9757. DOI: 10.1613/jair.1141. URL: https://jair.org/index.php/jair/article/view/10362 (cit. on p. 6).

[81] New Jun Jie, Wu Yujin. "Structured Multi-Agent World Models". In: () (cit. on pp. 10, 11, 23).

[82] R Nissim and R Brafman. "Distributed heuristic forward search for multi-agent planning". In: *J. Artif. Intell. Res.* 51 (Oct. 2014), pp. 293–332. ISSN: 1076-9757,1943-5037. DOI: 10.1613/jair.4295. URL: https://jair.org/index.php/jair/article/view/10909 (cit. on p. 6).

[83] Raz Nissim and R Brafman. "Multi-agent A* for parallel and distributed systems". In: *Adapt Agent Multi-agent Syst* (June 2012), pp. 1265–1266 (cit. on p. 6).

[84] R Olfati-Saber. "Distributed Kalman filtering for sensor networks". In: *Conference on Decision and Control* 12 (Dec. 2007), pp. 5492–5498. DOI: 10.1109/CDC.2007.4434303. URL: http://ieeexplore.ieee.org/document/4434303/ (cit. on p. 5).

[85] Minting Pan et al. "Iso-Dream: Isolating and Leveraging Noncontrollable Visual Dynamics in World Models". In: *arXiv [cs.LG]* (May 2022). arXiv: 2205.13817 [cs.LG] (cit. on pp. 7, 8, 11, 14, 15).

[86] Young Joon Park, Yoon Sang Cho, and Seoung Bum Kim. "Multi-agent reinforcement learning with approximate model learning for competitive games". en. In: *PLoS One* 14.9 (Sept. 2019), e0222215. ISSN: 1932-6203 (cit. on pp. 10, 13, 15, 17, 18, 21, 23).

[87] Tim Pearce et al. "Scaling laws for pre-training agents and world models". In: *arXiv [cs.LG]* (Nov. 2024). arXiv: 2411.04434 [cs.LG]. URL: http://arxiv.org/abs/2411.04434 (cit. on p. 23).

[88] A Peddemors and Eiko Yoneki. "Decentralized probabilistic world modeling with cooperative sensing". In: *Electron Commun Eur Assoc Softw Sci Technol* 17 (Feb. 2009). DOI: 10.14279/tuj.eceasst.17.214. URL: http://ubsrvweb09.ub.tu-berlin.de/eceasst/article/view/214 (cit. on pp. 4, 6, 21).

[89] F Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 22).

[90] Arnu Pretorius et al. "Learning to communicate through imagination with model-based deep multi-agent reinforcement learning". In: *https://openreview.net › forumhttps://openreview.net › forum* (Oct. 2020) (cit. on pp. 10, 12–14, 17, 19–21, 23).

[91] Tabish Rashid et al. "QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning". In: *arXiv [cs.LG]* (Mar. 2018). arXiv: 1803.11485 [cs.LG]. URL: http://arxiv.org/abs/1803.11485 (cit. on pp. 11, 16, 18–20).

[92] Craig W Reynolds. "Flocks, herds and schools: A distributed behavioral model". In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, Aug. 1987. ISBN: 9780897912273 (cit. on p. 2).

[93] Reuven Y Rubinstein. "Cross-entropy and rare events for maximal cut and partition problems". en. In: *ACM Trans. Model. Comput. Simul.* 12.1 (Jan. 2002), pp. 27–53. ISSN: 1049-3301,1558-1195. DOI: 10.1145/511442.511444. URL: https://dl.acm.org/doi/abs/10.1145/511442.511444 (cit. on p. 19).

[94] Stuart J Russell and Peter Norvig. *Artificial intelligence: A modern approach*. Prentice Hall, Jan. 2010. ISBN: 9781282645929 (cit. on p. 3).

[95] Mikayel Samvelyan et al. "The StarCraft Multi-Agent Challenge". In: *arXiv [cs.LG]* (Feb. 2019). arXiv: 1902.04043 [cs.LG]. URL: http://arxiv.org/abs/1902.04043 (cit. on pp. 16–19).

[96] Julian Schrittwieser et al. "Mastering Atari, Go, chess and shogi by planning with a learned model". en. In: *Nature* 588.7839 (Dec. 2020), pp. 604–609. ISSN: 0028-0836 (cit. on pp. 4, 8, 20).

[97] Pier Giuseppe Sessa, Maryam Kamgarpour, and Andreas Krause. "Efficient model-based multi-agent Reinforcement Learning via optimistic equilibrium computation". en. In: *International Conference on Machine Learning* (2022) (cit. on pp. 10, 13, 15, 17, 18, 21, 23).

[98] Wenling Shang et al. "Agent-Centric Representations for Multi-Agent Reinforcement Learning". en. In: *arXiv.org* abs/2104.09402 (Apr. 2021). ISSN: 2331-8422 (cit. on pp. 10, 12–14, 17, 19, 21).

[99] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". en. In: *Nature* 529.7587 (Jan. 2016), pp. 484–489. ISSN: 0028-0836. DOI: 10.1038/nature16961. URL: https://www.nature.com/articles/nature16961 (cit. on p. 8).

[100] Haolin Song et al. "MA2CL:Masked Attentive Contrastive Learning for Multi-Agent Reinforcement Learning". In: *arXiv [cs.LG]* (June 2023). arXiv: 2306.02006 [cs.LG] (cit. on pp. 10, 13, 15, 17, 19, 21, 23).

[101] Michal Stolba, Daniel Fiser, and Antonín Komenda. "Admissible Landmark Heuristic for Multi-Agent Planning". In: *Int Conf Autom Plan Sched* (Apr. 2015), pp. 211–219. DOI: 10.1609/icaps.v25i1.13719. URL: https://ojs.aaai.org/index.php/ICAPS/article/view/13719 (cit. on p. 6).

[102] Jayakumar Subramanian, Amit Sinha, and Aditya Mahajan. "Robustness and Sample Complexity of Model-Based MARL for General-Sum Markov Games". In: *Dyn. Games Appl.* 13.1 (Mar. 2023), pp. 56–88. ISSN: 2153-0785,2153-0793 (cit. on pp. 10, 17, 21).

[103] Ruixiang Sun et al. "Learning latent dynamic robust representations for world models". In: *arXiv [cs.LG]* (May 2024). arXiv: 2405.06263 [cs.LG]. URL: http://arxiv.org/abs/2405.06263 (cit. on pp. 7, 14).

[104] Rich Sutton. *The Bitter Lesson*. en. http://www.incompleteideas.net/IncIdeas/BitterLesson.html. Accessed: 2024-5-. Mar. 2019 (cit. on pp. 2, 3).

[105] Richard S Sutton. "Dyna, an integrated architecture for learning, planning, and reacting". In: *SIGART Bull.* 2.4 (July 1991), pp. 160–163. ISSN: 0163-5719. DOI: 10.1145/122344.122377. URL: https://doi.org/10.1145/122344.122377 (cit. on pp. 4, 5, 14, 16, 17).

[106] M Tambe and Weixiong Zhang. "Towards flexible teamwork in persistent teams". In: *Proceedings International Conference on Multi Agent Systems (Cat. No.98EX160)*. IEEE Comput. Soc, 2002. ISBN: 9780818685002. DOI: 10.1109/icmas.1998.699065. URL: http://dx.doi.org/10.1109/icmas.1998.699065 (cit. on p. 6).

[107] Amirhossein Tamjidi et al. "Efficient recursive distributed state estimation of hidden Markov models over unreliable networks". In: *Auton. Robots* 44.3 (Mar. 2020), pp. 321–338. ISSN: 0929-5593,1573-7527 (cit. on pp. 4, 6, 12).

[108] Alejandro Torreño, Eva Onaindia, and Óscar Sapena. "FMAP: Distributed Cooperative Multi-Agent Planning". In: *arXiv [cs.AI]* (Jan. 2015), pp. 606–626. DOI: 10.1007/s10489-014-0540-2. arXiv: 1501.07250 [cs.AI]. URL: http://link.springer.com/10.1007/s10489-014-0540-2 (cit. on p. 6).

[109] Alejandro Torreño et al. "Cooperative Multi-Agent Planning: A Survey". In: *ACM Comput. Surv.* 50.6 (Nov. 2017), pp. 1–32. ISSN: 0360-0300. DOI: 10.1145/3128584. URL: https://doi.org/10.1145/3128584 (cit. on pp. 4, 6).

[110] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I Guyon et al. Curran Associates, Inc., 2017, pp. 5998–6008 (cit. on pp. 10, 12, 16, 19, 23).

[111] Aravind Venugopal et al. "MABL: Bi-level latent-variable world model for sample-efficient multi-agent reinforcement learning". In: *Adapt Agent Multi-agent Syst* (Apr. 2023), pp. 1865–1873. DOI: 10.5555/3635637.3663049. eprint: 2304.06011. URL: https://www.ifaamas.org/Proceedings/aamas2024/pdfs/p1865.pdf (cit. on pp. 9, 11, 13, 14, 17, 19, 21–23).

[112] Rongxiao Wang et al. "MA-TDMPC: Multi-Agent Temporal Difference for Model Predictive Control". In: *2023 2nd International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM)*. IEEE, July 2023, pp. 256–266 (cit. on pp. 4, 10, 12, 13, 15–17, 19, 21–23).

[113] Rose Wang et al. "Model-based Reinforcement Learning for Decentralized Multiagent Rendezvous". In: *Proceedings of the 2020 Conference on Robot Learning*. Ed. by Jens Kober, Fabio Ramos, and Claire Tomlin. Vol. 155. Proceedings of Machine Learning Research. PMLR, 2021, pp. 711–725. URL: https://proceedings.mlr.press/v155/wang21d.html (cit. on pp. 10, 11, 13, 14, 19, 21, 23).

[114] Xihuai Wang, Zhicheng Zhang, and Weinan Zhang. "Model-based Multi-agent Reinforcement Learning: Recent Progress and Prospects". In: *arXiv [cs.MA]* (Mar. 2022). arXiv: 2203.10603 [cs.MA]. URL: http://arxiv.org/abs/2203.10603 (cit. on p. 33).

[115] Zhizun Wang and David Meger. "Leveraging world model disentanglement in value-based multi-agent reinforcement learning". In: *arXiv [cs.LG]* (Sept. 2023). arXiv: 2309.04615 [cs.LG]. URL: http://arxiv.org/abs/2309.04615 (cit. on pp. 10, 11, 13–17, 19–21).

[116] Manuel Watter et al. "Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images". In: *arXiv [cs.LG]* (June 2015). arXiv: 1506.07365 [cs.LG] (cit. on pp. 3, 7, 14).

[117] Daniël Willemsen, Mario Coppola, and Guido C H E de Croon. "MAMBPO: Sample-efficient multi-robot reinforcement learning using learned world models". In: *arXiv [cs.RO]* (Mar. 2021). arXiv: 2103.03662 [cs.RO] (cit. on pp. 10, 13, 14, 16, 17, 21–23).

[118] Jeffrey Wishart et al. *Fundamentals of connected and automated vehicles*. en. Warrendale: SAE International, Apr. 2022. ISBN: 9780768099812 (cit. on p. 1).

[119] Philipp Wu et al. "DayDreamer: World models for physical robot learning". In: *arXiv [cs.RO]* (June 2022). arXiv: 2206.14176 [cs.RO]. URL: http://arxiv.org/abs/2206.14176 (cit. on pp. 4, 7, 24).

[120] Zifan Wu et al. "Models as agents: Optimizing multi-step predictions of interactive local models in model-based multi-agent reinforcement learning". en. In: *Proc. Conf. AAAI Artif. Intell.* (2023). ISSN: 2159-5399 (cit. on pp. 8, 10, 12–17, 19–21, 23).

[121] Wenli Xiao, Yiwei Lyu, and John Dolan. "Model-based Dynamic Shielding for Safe and Efficient Multi-Agent Reinforcement Learning". In: *arXiv [cs.LG]* (Apr. 2023). arXiv: 2304.06281 [cs.LG]. URL: http://arxiv.org/abs/2304.06281 (cit. on pp. 13–15, 19–21, 23).

[122] Annie Xie et al. "Learning Latent Representations to Influence Multi-Agent Interaction". In: *Proceedings of the 2020 Conference on Robot Learning*. Ed. by Jens Kober, Fabio Ramos, and Claire Tomlin. Vol. 155. Proceedings of Machine Learning Research. PMLR, 2021, pp. 575–588 (cit. on pp. 10, 11, 13, 14, 17, 20, 21, 23).

[123] Xiaopeng Yu et al. "Model-based opponent modeling". In: *arXiv [cs.LG]* (Aug. 2021). arXiv: 2108.01843 [cs.LG]. URL: http://arxiv.org/abs/2108.01843 (cit. on pp. 10, 13, 17, 19–21).

[124] Hongxin Zhang et al. "Building Cooperative Embodied Agents Modularly with Large Language Models". In: *arXiv [cs.AI]* (July 2023). arXiv: 2307.02485 [cs.AI] (cit. on pp. 10, 12, 17, 19–21, 24).

[125] Hongxin Zhang et al. "COMBO: Compositional World Models for Embodied Multi-Agent Cooperation". In: *arXiv [cs.CV]* (Apr. 2024). arXiv: 2404.10775 [cs.CV]. URL: http://arxiv.org/abs/2404.10775 (cit. on pp. 10, 12–14, 17, 19–21, 24).

[126] Junbo Zhang and Kaisheng Ma. "Rethinking the augmentation module in contrastive learning: Learning hierarchical augmentation invariance with expanded views". In: *arXiv [cs.CV]* (June 2022). arXiv: 2206.00227 [cs.CV]. URL: http://arxiv.org/abs/2206.00227 (cit. on p. 16).

[127] Kaiqing Zhang et al. "Model-based multi-agent RL in zero-sum Markov games with near-optimal sample complexity". In: *J. Mach. Learn. Res.* 24.1 (Mar. 2024), pp. 8286–8338. ISSN: 1532-4435 (cit. on pp. 10, 11, 13, 21).

[128] Lunjun Zhang, Ge Yang, and Bradly C Stadie. "World Model as a Graph: Learning Latent Landmarks for Planning". en. In: *ICML* (2020) (cit. on pp. 8, 14, 15).

[129] Marvin Zhang et al. "SOLAR: Deep Structured Representations for Model-Based Reinforcement Learning". In: *arXiv [cs.LG]* (Aug. 2018). arXiv: 1808.09105 [cs.LG]. URL: http://arxiv.org/abs/1808.09105 (cit. on pp. 14, 15).

[130] Qizhen Zhang et al. "Centralized Model and Exploration Policy for Multi-Agent RL". In: *arXiv [cs.AI]* (July 2021). arXiv: 2107.06434 [cs.AI]. URL: http://arxiv.org/abs/2107.06434 (cit. on pp. 10, 13, 14, 16, 17, 21–23).

[131] Weinan Zhang et al. "Model-based Multi-agent Policy Optimization with Adaptive Opponent-wise Rollouts". In: *arXiv [cs.LG]* (May 2021). arXiv: 2105.03363 [cs.LG]. URL: http://arxiv.org/abs/2105.03363 (cit. on pp. 10, 12, 17, 20, 21).

[132] Yang Zhang et al. "Decentralized transformers with centralized aggregation are sample-efficient multi-agent world models". In: *arXiv [cs.LG]* (June 2024). arXiv: 2406.15836 [cs.LG]. URL: http://arxiv.org/abs/2406.15836 (cit. on pp. 10, 12, 13, 17, 21).

[133] Zhuoya Zhao et al. "A Brain-inspired Theory of Collective Mind Model for Efficient Social Cooperation". In: *IEEE Transactions on Artificial Intelligence* PP.99 (2024), pp. 1–10. ISSN: 2691-4581 (cit. on pp. 12–15, 19, 23).

[134] Gaoyue Zhou et al. "DINO-WM: World models on pre-trained visual features enable zero-shot planning". In: *arXiv [cs.RO]* (Nov. 2024). arXiv: 2411.04983 [cs.RO]. URL: http://arxiv.org/abs/2411.04983 (cit. on p. 19).

[135] Jie Zhou et al. "Graph neural networks: A review of methods and applications". In: *arXiv [cs.LG]* (Dec. 2018). arXiv: 1812.08434 [cs.LG]. URL: http://arxiv.org/abs/1812.08434 (cit. on pp. 7, 11).

[136] Changxi Zhu, Mehdi Dastani, and Shihan Wang. "A Survey of Multi-Agent Reinforcement Learning with Communication". In: *arXiv [cs.MA]* (Mar. 2022). arXiv: 2203.08975 [cs.MA]. URL: http://arxiv.org/abs/2203.08975 (cit. on p. 33).

[137] Zheng Zhu et al. "Is Sora a world simulator? A comprehensive survey on General world models and beyond". In: *arXiv [cs.CV]* (May 2024). arXiv: 2405.03520 [cs.CV]. URL: http://arxiv.org/abs/2405.03520 (cit. on p. 33).

[138] Danping Zou, Ping Tan, and Wenxian Yu. "Collaborative visual SLAM for multiple agents:A brief survey". en. In: *Virtual Reality & Intelligent Hardware* 1.5 (Oct. 2019), pp. 461–482. ISSN: 2096-5796. DOI: 10.1016/j.vrih.2019.09.002. URL: https://www.sciencedirect.com/science/article/pii/S2096579619300634 (cit. on pp. 3, 4, 6).

[139] Michal Štolba and Antonín Komenda. "Relaxation heuristics for multiagent planning". In: *Proceedings of the 24th international conference on automated planning and scheduling (ICAPS'14)*. 2014, pp. 298–306 (cit. on p. 6).

[140] Michal Štolba, Jan Tožička, and Komenda Antonín. "Quantifying privacy leakage in multi-agent planning". In: *Proceedings of the 4rd ICAPS workshop on distributed and multi-agent planning (DMAP'16)*. 2016, pp. 80–88 (cit. on p. 6).

A Survey Details

A.1 Survey Approach

This survey employed a systematic methodology to explore the landscape of MAWMs. Searches were conducted on Google, Google Scholar, and arXiv using targeted terms such as "Decentralized World Models," "Centralized World Models," "Multi-Agent World Models," "Collaborative World Models," and "Model-Based Multi-Agent Reinforcement Learning." To ensure comprehensiveness, a bi-directional citation analysis was performed, tracing both references cited within relevant works and subsequent studies that cited these sources. To capture the evolution of ideas, the survey intentionally spanned a wide temporal scope, extending to the early foundational studies. A taxonomy of MAWMs was developed to classify studies based on their characteristics and contributions, providing a structured comparison framework. The comprehensive process facilitated the identification of foundational works, significant advancements, and emerging trends. This approach not only highlights the historical progression of the field but also reveals gaps and opportunities for future research, offering a complete and well-rounded perspective on MAWMs.

A.2 Related Surveys

This survey provides the first comprehensive examination of MAWMs. While Wang et al. [114] provides a review of model-based MARL methods, it has a much more limited scope, rather than this work's broader framework for world modeling. While Luo et al. [71] includes a brief section on model-based MARL, it does not provide a comprehensive framework, nor does it fully address the broader challenges of multi-agent world modeling. Guan et al. [34] examines world models in autonomous driving but focuses on single-vehicle driving scenario generation, planning and control, environmental modeling, and trajectory prediction, rather than multi-agent coordination. Other surveys examine aspects of MARL [33, 136] and model-based RL [77], but none specifically address decentralized architectures and multi-agent world modeling. Finally, Zhu et al. [137] explores world models primarily through the lens of LVMs as implicit world models, with only limited discussion of multi-agent scenarios.

A.3 Generative AI Usage Statement

This manuscript was developed with assistance from ChatGPT (4o, o1-mini, and o1) and Claude (version Sonnet 3.5). Claude was used in the following ways:

- Initial generation of Tikz/Forest diagram syntax, with significant author modification on the semantics;
- Drafting and editing suggestions for clarity, conciseness, and technical accuracy; and
- Basic copy-editing suggestions for grammar, formatting, and consistent style.

All generated content was thoroughly reviewed, verified, and edited by the authors to ensure accuracy and originality. Key technical contributions, analyses, and conclusions represent the authors' novel intellectual work. The complete chat logs and prompts are available upon request.

B Acronyms

Acronyms

ACNN Agent-Centric Neural Network. 10, 13, 17

AI Artificial Intelligence. 5

AMLAPN Approximate Model Learning using Auxiliary Prediction Networks. 10, 13, 17

AORPO Adaptive Opponent-wise Rollout Policy Optimization. 10, 17, 20

AWPO Advantage-Weighted Policy Optimization. 18, 20

BYOL Bootstrap Your Own Latent space. 15, 16

C-MAWM Centralized Multi-Agent World Model. 9, 10, 23

C-SWM Contrastively-trained Structured World Model. 7

CEM Cross Entropy Method. 7, 19

CoELA Cooperative Embodied Language Agent. 10, 17, 20, 24

COMBO Compositional wOrld Model-based emBOdied. 10, 13, 17, 20, 24

CPS Cooperative Prioritized Sweeping. 17, 18

CTDE Centralized Training, Decentralized Execution. 9–11, 17

CVAE Conditional Variational AutoEncoder. 14

D-MAWM Decentralized Multi-Agent World Model. 9, 11, 12, 23

DCOP Distributed Constraint Optimization Problem. 4

DDN Dynamic Decision Network. 18

Dec-POMDP Decentralized Partially Observable Markov Decision Process. 10

DINO-WM DINO World Model. 8

DSSM Deep State Space Model. 10, 11, 13, 17, 20

E2C Embed to Control. 7

GAIA-1 Generative AI for Autonomy-1. 8, 24

GenAI Generative AI. 8

GNN Graph Neural Network. 7, 11, 23

GP Gaussian Process. 5

H-MARL Hallucinated Multi-Agent Reinforcement Learning. 13, 17, 18

HMM Hidden Markov Model. 4, 5

HPP Hierarchical Predictive Planning. 10, 13

HRSSM Hybrid Recurrent State Space Model. 7

HTN Hierarchical Task Network. 5

ICF Iterative Conservative Fusion. 6

IRIS Imagination with auto-Regression over an Inner Speech. 8

IS Intention Sharing. 10, 13, 17, 20

JEPA Joint Embedded Predictive Architecture. 4, 8

KL Kullback-Leibler. 7

L^{3}P Learning Latent Landmarks for Planning. 8

LDS Linear Dynamical System. 7

LILI Learning and Influencing Latent Intent. 10, 13, 17

LLM Large Language Model. 8, 17, 19, 20, 24

LQR Linear Quadratic Regulator. 7

LTL Linear Temporal Logic. 20, 23

LVM Large Vision Model. 17, 20, 33

M-PPO Multi-Agent PPO. 17, 18

M-QMIX Masked reconstruction task with QMIX. 10, 11, 13, 16, 17, 19, 23

MA-MuJuCo Multi-Agent MuJoCo. 16, 18

MA-SPL MA-Self-Predictive Learning. 14, 15

MA-TDMPC Multi-Agent Temporal Difference MPC. 4, 10, 12, 13, 17, 19, 22, 23

MA-TDR MA-Transition Dynamics Reconstruction. 15

MA2CL Multi-Agent Masked Attentive Contrastive Learning. 10, 13, 15, 17, 19

MABL Multi-Agent Bi-Level world model. 11, 13, 17, 19, 22

MACD Multi-Agent Counterfactual Dreamer. 10, 13, 16–18, 22

MACI Multi-Agent Communication through Imagination. 10, 13, 17

MAG Models as AGents. 10, 12–14, 17, 20

MAMBA Multi-Agent Model-Based Approach. 4, 10, 12, 13, 16, 17, 22, 23

MAMBPO Multi-Agent Model-Based Policy Optimization. 13, 14, 17, 22

MAPO-LSO Multi-Agent Policy Optimization with Latent Space Optimization. 10, 12–15, 17, 22, 23

MARCO Multi-Agent RL with Centralized mOdels and exploration. 10, 13, 17, 22

MARIE Multi-Agent auto-Regressive Imagination for Efficient learning. 10, 12, 13, 17

MARL Multi-Agent Reinforcement Learning. 9, 10, 15, 16, 18–20, 33

MAS Multi-Agent System. 1–3, 5, 9, 16, 18, 20, 24

MAT Multi-Agent Transformer. 15

MAWM Multi-Agent World Model. 1, 2, 4, 5, 8, 9, 13, 14, 16–24, 33

MAZero Multi-Agent MuZero. 4, 10, 13, 17, 18, 20, 22, 23

MB-MARL Model-Based Multi-Agent Reinforcement Learning. 10

MBCA Model-Based Credit Assignment. 18

MBDS Model-Based Dynamic Shielding. 13, 20

MBOM Model-Based Opponent Modeling. 13, 17, 20

MCTS Monte-Carlo Tree Search. 8, 17–20

MDP Markov Decision Process. 18

MPC Model-Predictive Control. 8, 14, 17, 19, 20

NeRF Neural Radiance Fields. 24

NEXP Non-Deterministic Exponential Time. 10

NN Neural Network. 6, 14, 24

OM Opponent Modeling. 9, 11, 16, 17, 20

PDDL Planning Domain Definition Language. 5

PILCO Probabilistic Inference for Learning COntrol. 5

PlaNet deep Planning Network. 7

Q-Mix Q-Mix. 11, 16, 18–20

RL Reinforcement Learning. 5, 7, 11, 13, 14, 17, 18, 20, 22, 33

RNN Recurrent Neural Network. 3, 10, 19

RSSM Recurrent State Space Model. 7

SLAM Simultaneous Localization and Mapping. 2, 4, 6

SMAC StarCraft Multi-Agent Challenge. 16–19

SMAWM Structured Multi-Agent World Models. 10, 11

SOLAR Stochastic Optimal control with LAtent Representations. 7

STRIPS Stanford Research Institute Problem Solver. 5

TD-MPC Temporal Difference Model Predictive Control. 4, 8, 14

TD-MPC2 Temporal Difference Model Predictive Control 2. 8, 14

TESSERACT Tensorised Actors. 10, 17, 18

TIMAR Transition-Informed Multi-Agent Representations. 10, 13, 15, 17, 19

TSM Temporal Segment Model. 10, 13, 17, 20

UCT Upper Confidence-bound for Trees. 18

VAE Variational AutoEncoder. 7, 16, 20

VDFD Value Decomposition Framework with Disentangled World Model. 10, 11, 13, 16, 17, 20

ViT Vision Transformer. 8

WM World Model. 1–20, 22–24