MAGIC: MULTI-AGENT GENERATIVE INTENTION COORDINATION

David Huk[1,2], Oliver Hamelijnck[1], Dimitris Demiris[1],Theodoros Damoulas[1,2,3]
[1]AI Team, Unilink LTD
[2]Warwick University, Department of Statistics
[3]Warwick University, Department of Computer Science
David.Huk@warwick.ac.uk

ABSTRACT

Multi-agent imitation learning (**MAIL**) aims to learn coordinated policies from expert demonstrations, but learning *dependent* multi-agent policies is often infeasible due to the combinatorial complexity of joint action modelling. As a result, many practical approaches assume a factorisation across agents, sacrificing expressivity and ignoring coordination present in expert data. We introduce **MAGIC**, a scalable framework for **M**ulti-**A**gent **G**enerative **I**ntention **C**oordination that induces a structured joint policy without explicitly modelling the high-dimensional joint action space. **MAGIC** follows a divide-and-conquer strategy: (i) we learn lightweight independent policies, (ii) we compress each agent's action distribution into a one-dimensional latent *intention* via either ρ- or Hilbert-space projections, and (iii) we learn a dependent generative model over intentions using diffusion-based copulas. This yields a scalable generative representation of the joint policy, enabling coordinated action sampling while preserving inter-agent dependencies. Moreover, in the centralised training with decentralised execution setting, **MAGIC** supports coordinated execution without communication by allowing intention values to be precomputed offline. We show that **MAGIC** out-performs **MAIL** baseline on challenging real-world dataset.

1 MOTIVATION

Multi-agent imitation learning (**MAIL**) seeks to learn coordinated policies from expert demonstrations. By avoiding the need for reward functions, **MAIL** can exploit the growing availability of large-scale observational data of coordinated behaviour. However, learning dependent multi-agent policies remains difficult due to the curse of dimensionality; representing a joint policy requires modelling complex dependencies among many agents with large action spaces. A central challenge in **MAIL** is:

How can we model coordination among many agents in a scalable way?

Existing approaches largely sidestep this challenge by assuming a factorisation across agents, reducing the problem to learning independent single-agent policies (Li et al., 2025). This assumption underlies both general MAIL and the popular paradigm of centralised training with decentralised execution (CTDE) (Amato, 2024), where agents must act independently at deployment. While factorisation enables scalability, it fundamentally limits expressivity as inter-agent dependencies present in expert demonstrations are ignored, leading to inconsistent and uncoordinated behaviour (Wang et al., 2021).

We introduce **MAGIC**, a scalable framework for **M**ulti-**A**gent **G**enerative **I**ntention **C**oordination. **MAGIC** overcomes the fully factorised assumption prevalent in **MAIL** by capturing dependency between agents' *intentions* - a latent variable that governs the actions taken by an agent. By compressing each agents action space, **MAGIC** captures inter-agent dependence while preserving scalability. Our approach follows a divide-and-conquer strategy: we start from independent per-agent policies, which are compressed into a single scalar, an agent's intention, and learn a full dependent generative model over these intentions.

This addresses the central challenge in **MAIL** by inducing a structured joint policy without requiring high-dimensional joint action modelling. Crucially, in the CTDE setting, **MAGIC** enables coordinated

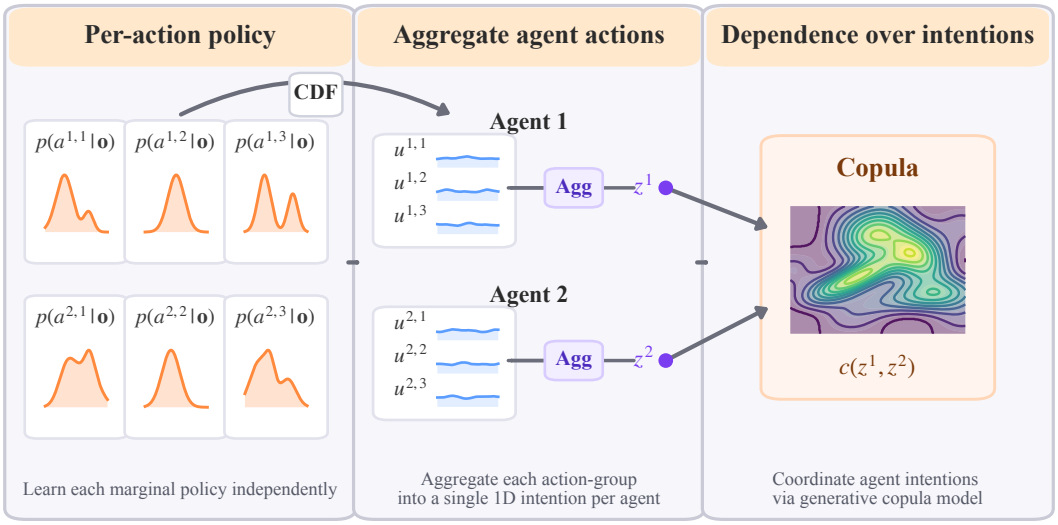

Figure 1: Overview of proposed **MAGIC** framework. Action marginals are modelled independently and are aggregated through a ρ or Hilbert transform into an intent variable z^i. A copula is placed over the intent variables, which induces correlation between the agents actions.

execution by endowing each agent with a set of precomputed intention values, allowing dependent behaviour without communication. In summary, we: (1) introduce **MAGIC**, a scalable dependent framework for **MAIL** and CTDE; (2) theoretically show that **MAGIC** can represent the true joint policy even under inter-agent dependence (3) propose two practical instantiations based on ρ and Hilbert space projections that effectively compress each agent's actions into an intention with dependence modelled with diffusion based copulas; and (4) demonstrate **MAGIC** on a challenging real-world dataset, outperforming state-of-the-art baselines.

2 Background

2.1 Multi-Agent Imitation Learning

Assume that at times $t = 1, \ldots, T$, we observe a collection of expert demonstrations of n agents denoted by $\boldsymbol{a}_t = (\boldsymbol{a}_t^1, \ldots, \boldsymbol{a}_t^n)$, each taking d-dimensional actions $\boldsymbol{a}_t^i = (a_t^{i,1}, \ldots, a_t^{i,d}) \in \mathbb{R}^d$ conditioned on state observations $\boldsymbol{o}_t^i \subseteq \mathbb{R}^s$. The *joint policy* over the n agents is a probability density function over the possible actions jointly taken by agents given their state:

$$\textbf{Joint policy:} \quad \boldsymbol{p}(\boldsymbol{a}_t^1, \ldots, \boldsymbol{a}_t^n \,|\, \boldsymbol{o}_t^1, \ldots, \boldsymbol{o}_t^n) : \mathbb{R}^{n \cdot d} \times \mathbb{R}^r \to \mathbb{R}^+, \quad r \leq n \cdot s, \tag{1}$$

where r accounts for possible redundancies in the state. The task of **MAIL** consists of learning a model for the joint policy in (1) to imitate the expert's demonstration. In the CTDE setting each agent must act solely on its local observation $\boldsymbol{o}_t^i$ at execution time, without communication with other agents. This induces decentralised marginal policies of the form $p^i(\boldsymbol{a}^i t + 1 \,|\, \boldsymbol{o}_t^i) : \mathbb{R}^d \times \mathbb{R}^s \to \mathbb{R}^+$. Throughout, $\hat{p}$ denotes estimated policies with superscripts for dimensions and subscripts for time.

2.2 Simple Factorisations and Beyond

The high-dimensional nature of the joint policy in (1) poses a fundamental modelling challenge due to the curse of dimensionality. A simple way to mitigate this is to impose strong independence assumptions by modelling each agent and each action dimension independently

$$\textbf{Independent policy:} \quad \boldsymbol{p}(\boldsymbol{a}_t \,|\, \boldsymbol{o}_t) \approx \prod_{i=1}^{n} \prod_{j=1}^{d} \widehat{p}^{\,i,j}(a_t^{i,j} \,|\, \boldsymbol{o}_t^i). \tag{2}$$

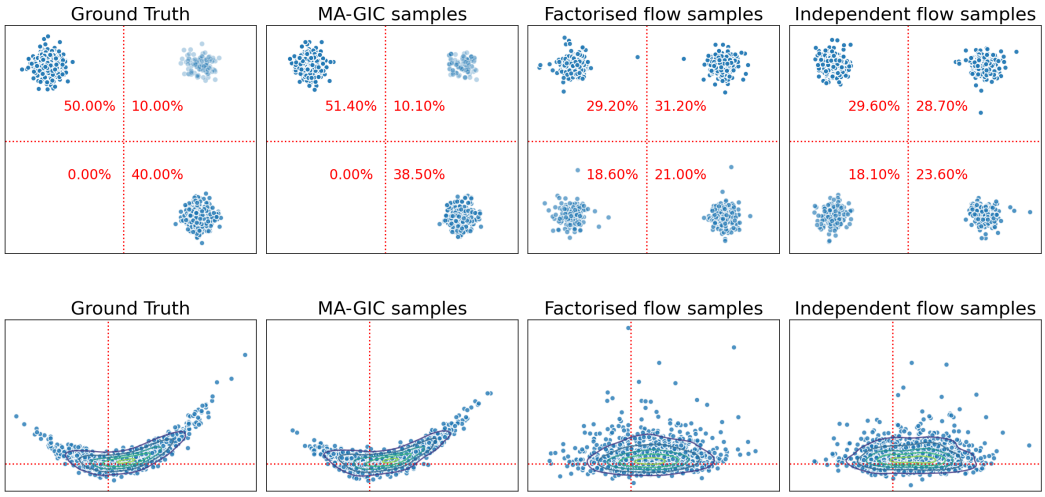

Figure 2: **Dependent Policy Imitation**: An illustrative example of a dependent policy in 2D, with one agent per coordinate, following Li et al. (2025). Independent and factorised policy models fail to capture the underlying dependence structure, whereas our method successfully recovers it.

While scalable, this policy neglects interactions among the d actions of each agent, and between the n agents themselves, yielding a fully factorised model that ignores joint structure. Instead, a more compelling approach is to assume independence across agents only

$$\textbf{Factorised policy:} \quad \boldsymbol{p}(\boldsymbol{a}_t \mid \boldsymbol{o}_t) \approx \prod_{i=1}^{n} \widehat{p}^{\,i}(\boldsymbol{a}_t^i \mid \boldsymbol{o}_t^i) \tag{3}$$

which may retain dependence between an agent's action dimensions. This assumption underlies many MAIL methods and recent work by Li et al. (2025); Lee et al. (2025) further improves on this with

$$\textbf{Projected policy:} \quad \boldsymbol{p}(\boldsymbol{a}_t \mid \boldsymbol{o}_t) \approx \arg\min_{\widehat{\boldsymbol{p}} \in \mathcal{P}} \mathcal{L}(\widehat{\boldsymbol{p}} \,||\, \boldsymbol{p}), \quad \text{with} \quad \mathcal{P} = \{\widehat{\boldsymbol{p}} : \widehat{\boldsymbol{p}}(\boldsymbol{a} \mid \boldsymbol{o}) = \prod_{i=1}^{n} \widehat{p}^{\,i}(\boldsymbol{a}^i \mid \boldsymbol{o}^i)\} \tag{4}$$

which is a projection of the true joint policy $\boldsymbol{p}$ onto the restricted model class of factorised policies $\mathcal{P}$ under the chosen loss $\mathcal{L}$. While all these approaches are compatible with the CTDE setting, they fundamentally fail to capture dependencies *between* agents, and therefore struggle to express the coordinated behaviour present in expert demonstrations.

3 Introducing Dependence Despite Decentralised Execution

The central challenge of MAIL under the CTDE paradigm is that, at execution time, agents must act independently without access to shared information or communication. We address this by representing the joint policy through a copula decomposition, which separates inter-agent dependencies from individual action distributions. In this formulation, the marginal policies and the copula capturing their dependence can be learned entirely during centralized training. Assuming a state-invariant copula makes the dependence time-invariant, allowing copula samples (quantiles) to be precomputed offline. Each agent then maps its assigned quantiles through its own state-conditioned marginal policy, enabling coordinated yet fully decentralized execution.

A *copula* is a multivariate distribution that uniquely captures dependence Joe (2014). Applying Sklar's thm. Sklar (1959) to the joint policy, the dependence structure is decoupled from the marginals:

Theorem 3.1 (Joint policy copula decomposition)**.** *Let $\boldsymbol{p}(\boldsymbol{a} \mid \boldsymbol{o})$ be a $(n \cdot d)$-dimensional joint policy function with absolutely continuous marginal policies $p^{i,j}(a^{i,j} \mid \boldsymbol{o})$ for $i = 1, \ldots, n$, $j = 1, \ldots, d$.*

Then there exists a copula with density function $\boldsymbol{c}(. \mid \boldsymbol{o}) : [0,1]^{n \cdot d} \times \mathbb{R}^r \to \mathbb{R}^+$ *such that* $\forall \boldsymbol{a} \in \mathbb{R}^{n \cdot d}$*:*

$$\boldsymbol{p}(\boldsymbol{a} \mid \boldsymbol{o}) = \prod_{i=1}^{n} \prod_{j=1}^{d} \left\{ p^{i,j}(a^{i,j} \mid \boldsymbol{o}) \right\} \cdot \boldsymbol{c}\big(P^{1,1}(a^{1,1} \mid \boldsymbol{o}), \ldots, P^{n,d}(a^{n,d} \mid \boldsymbol{o}) \,\big|\, \boldsymbol{o}\big) \tag{5}$$

where $P^{i,j}$ *are the cumulative distribution functions (CDFs) of marginal policies.*

The copula is a distribution on the quantiles of all action dimensions (defined on $[0,1]^{n \cdot d}$), which encodes dependence between actions. This can best be understood via the two-step sampling mechanism corresponding to (5): (i) Sample $(n \cdot d)$-dimensional quantiles $\mathbf{u} \sim \boldsymbol{c}(. \mid \boldsymbol{o})$; (ii) Using inverse probability sampling with inverse marginal policy CDFs to transform quantiles into actions $a^{i,j} = {P^{i,j}}^{-1}(u^{i,j} \mid \boldsymbol{o})$. Thus, the marginals are responsible for adapting the next action's distribution to the observed state, while the copula selects quantiles that are likely to occur together.

As is common in the CTDE framework, we make the following assumptions:

Assumption 1. *Policies* $p^{i,j}$ *only depend on a subset of observations* $\boldsymbol{o}^i \subseteq \boldsymbol{o}$ *available to the agent.*

Assumption 2. *The copula density* $\boldsymbol{c}$ *of the joint policy is state-invariant:* $\boldsymbol{c}(\cdot \mid \boldsymbol{o}) = \boldsymbol{c}(\cdot)$ *for all* $\boldsymbol{o}$*.*

The first assumption ensures that each agent has enough information available to define their policy. The second assumption simplifies the dependence among actions and will enable decentralised execution. This is reasonable in settings where coordination patterns (e.g formations) persist across states. Under Assumptions 1 and 2, we propose the copula policy model:

$$\textbf{Copula policy: } \boldsymbol{p}(\boldsymbol{a}_t \mid \boldsymbol{o}_t) \approx \Big(\prod_{i=1}^{n} \prod_{j=1}^{d} \widehat{p}^{\,i,j}(a_t^{i,j} \mid \boldsymbol{o}_t^i) \Big) \cdot \widehat{\boldsymbol{c}}\big(\widehat{P}^{1,1}(a_t^{1,1} \mid \boldsymbol{o}_t^1), \ldots, \widehat{P}^{n,d}(a_t^{n,d} \mid \boldsymbol{o}_t^n)\big). \tag{6}$$

Finally, we provide sufficient and necessary conditions for each of the aforementioned policy models to hold only in terms of the dependence structure of the joint policy.

Corollary 3.2. *Under Assumptions 1 and 2, for* $\boldsymbol{u} = (\boldsymbol{u}^1, \ldots, \boldsymbol{u}^n) \in [0,1]^{n \cdot d}$*, the following holds:*

1. $\boldsymbol{p}(\boldsymbol{a}|\boldsymbol{o})$ *admits an independent (2) policy* $\iff$ $\boldsymbol{c}(\boldsymbol{u}) = 1$ *a.e. in* $\boldsymbol{u}$*.*
2. $\boldsymbol{p}(\boldsymbol{a}|\boldsymbol{o})$ *admits a factorised (3) or projected (4) policy* $\iff$ $\boldsymbol{c}(\boldsymbol{u}) = \prod_{i=1}^{n} c^i(\boldsymbol{u}^i)$ *a.e. in* $\boldsymbol{u}$*.*
3. $\boldsymbol{p}(\boldsymbol{a}|\boldsymbol{o})$ *always admits a copula policy representation (6)* [1]*.*

Proof. Apply Theorem 3.1 on the joint, factorised and projected policies. Then group terms into a product of all $n \cdot d$ action marginals as in (2) and the remaining copula(s). Finally, identify terms appearing in both the true joint policy and each policy model to conclude. □

This result describes **MAIL** strictly in terms of the dependence between agents. If either point 1 or 2 is satisfied, then **MAIL** is nothing but a collection of $n \cdot d$ or n single-agent problems, respectively. In contrast, the joint policy *always* admits a copula, and so the copula policy remains expressive enough to recover the true joint policy. We provide an illustrative example in Figure 2 .

Dependent Decentralised Execution. The copula policy in (6) generates coordinated actions by first sampling joint quantiles from the copula and then mapping them through each agent's marginal policy. This can be implemented under the CTDE setting as follows. During training, we (i) fit marginal policies $\widehat{p}^{\,i,j}(a_t^{i,j} \mid \boldsymbol{o}_t^i)$ independently, (ii) estimate the copula over the resulting quantiles

$$\boldsymbol{u}_t = \big(\widehat{P}^{1,1}(a_t^{1,1} \mid \boldsymbol{o}_t^1), \ldots, \widehat{P}^{n,d}(a_t^{n,d} \mid \boldsymbol{o}_t^n)\big), \qquad t = 1, \ldots, T.$$

After training, we draw M i.i.d. samples $\{\boldsymbol{u}^{(m)}\}_{m=1}^{M} \sim \widehat{\boldsymbol{c}}$ and store them offline. At execution time, agent i is assigned its corresponding quantiles $\{\boldsymbol{u}^{(m),i}\}_{m=1}^{M}$, and all agents iterate through the same sample index m using a shared schedule (e.g., sequentially or via a common random seed). Each agent produces its action through inverse-CDF sampling using its local marginal. This preserves decentralised execution while inducing coordinated, statistically dependent actions.

[1] The copula policy can recover an independent policy by setting $\boldsymbol{c}(\boldsymbol{u}) = 1$, and a factorised or projected policy by setting $\boldsymbol{c}(\boldsymbol{u}) = \prod_{i=1}^{n} c^i(\boldsymbol{u}^i)$, therefore generalising these policy models.

4 GENERATIVE COORDINATION MODEL – MAGIC

While a copula policy (6) can capture the joint policy under Assumptions 1 and 2, the curse of dimensionality still poses an issue as the copula is a potentially complex and multimodal $(n \cdot d)$-dimensional density. To combat this, we propose an approach for **M**ulti-**A**gent **G**enerative **I**ntention **C**oordination called **MAGIC** as a three-part divide-and-conquer solution to modelling the coordination of the $(n \cdot d)$-dimensional copula. In Section 4.1, we estimate individual actions and extract their underlying coordination patterns. These are then aggregated into lower-dimensional, per-player intentions in Section 4.2, which we ultimately coordinate using a generative model in Section 4.3.

4.1 PART I: INDEPENDENT ACTION LEARNING

First, marginal policies from 6 are learnt by fitting independent conditional densities to each of the $n \cdot d$ action dimensions. Each marginal $\widehat{p}^{i,j}_{\theta^{i,j}}(a^{i,j} \mid \boldsymbol{o}^i)$ is parameterised by a lightweight network $N^{i,j}_{\theta^{i,j}}(\boldsymbol{o}^i)$ (Sec. B) and trained on observations with a Continuous Ranked Probability Score (CRPS)

$$\text{CRPS}\big(\widehat{P}^{i,j}_{\theta^{i,j}}, a^{i,j}_t\big) = \int_{-\infty}^{\infty} \big[\widehat{P}^{i,j}_{\theta^{i,j}}(z|\boldsymbol{o}^i) - \mathbf{1}_{\{a^{i,j}_t \geq z\}}\big]^2 \, dz, \tag{7}$$

where $\mathbf{1}$ represents the indicator function with $\mathbf{1}_A = 1$ if A is true and is 0 otherwise (Gneiting & Raftery, 2007). As the CRPS is a strictly proper scoring rule, minimising it recovers the parameters $\theta^{i,j}$ such that $\widehat{p}^{i,j}_{\theta^{i,j}}$ matches the true data-generating process within the model family. This estimation can be trivially parallelised across the $(n \cdot d)$ marginals to speed up computations.

We then use the trained marginal policy CDFs to obtain observed quantiles via

$$\boldsymbol{u}_t = \big(\widehat{P}^{1,1}_{\theta^{1,1}}(a^{1,1}_t|\boldsymbol{o}^1_t), \ldots, (\widehat{P}^{n,d}_{\theta^{n,d}}(a^{n,d}_t|\boldsymbol{o}^n_t)\big). \tag{8}$$

These quantiles $\{\boldsymbol{u}_t\}_{t=1}^T$ serve as observations from the copula $\boldsymbol{c}(.)$ coordinating the $n \cdot d$ actions. This training procedure is best practice for copula models (Hofert et al., 2018) and ensures the validity of the marginals and the validity of training a copula on the resulting quantiles (Ashok et al., 2024).

4.2 PART II: INTENTION AS DEPENDENCE AGGREGATION

To alleviate the burden of modelling the full $(n \cdot d)$-dimensional copula, we compress the within-agent dependence structure into a single latent variable per agent, which we term its *intention*. For the action quantiles

$$\boldsymbol{u}^i = (\widehat{P}^{i,1}(a^{i,1}_t \mid \boldsymbol{o}^i_t), \ldots, \widehat{P}^{i,d}(a^{i,d}_t \mid \boldsymbol{o}^i_t)) \in [0,1]^d,$$

we define an aggregation map

$$Z : [0,1]^d \to [0,1],$$

and denote the resulting intention by $z^i = Z(\boldsymbol{u}^i)$. The map Z is constructed so that $z^i \sim U[0,1]$, ensuring that the vector of intentions lies in the copula domain and can be coordinated independently of marginal effects. Below, we introduce candidate aggregation mechanisms, each encoding a different notion of within-agent dependence.

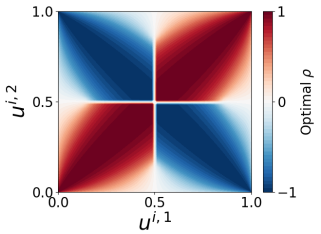

Figure 3: **The ρ Aggregation Map.** Compute the pairwise correlation for each quantiles $\boldsymbol{u}^i \in [0,1]^d$.

Aggregation via ρ. For clarity, we describe the construction for $d = 2$; extensions to $d > 2$ follow analogously by taking a summary of the correlation matrix (e.g a mean). We define the intention of agent i as the Gaussian copula correlation parameter that best explains its quantiles $\boldsymbol{u}^i$. Specifically, letting Φ^{-1} denote the element-wise standard normal transform, we compute

$$\rho^i = \underset{\rho \in [-1,1]}{\arg\max}\, \mathcal{N}\left(\Phi^{-1}(\boldsymbol{u}^i); \mathbf{0}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right), \tag{9}$$

i.e., the Gaussian copula parameter maximising the likelihood of the transformed quantiles. This compresses the d-dimensional within-agent dependence into a single scalar $\rho^i \in [-1,1]$, which we subsequently map to $z^i \in [0,1]$ via its empirical CDF to ensure uniform marginals.

The resulting intention encodes concordance across the agent's actions: values near 1 indicate strong positive coordination (high quantiles co-occur), values near 0 indicate strong negative coordination, and values near 0.5 correspond to approximate independence.

During decentralised execution, a precomputed intention z^i is first mapped back to ρ^i via the inverse ECDF. We then generate coordinated action quantiles by sampling

$$\boldsymbol{u}^i = \Phi(\boldsymbol{x}), \qquad \boldsymbol{x} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 1 & \rho^i \\ \rho^i & 1 \end{bmatrix}\right). \tag{10}$$

Aggregation via Hilbert Curves. We project the d-dimensional quantiles $\boldsymbol{u}^i \in [0,1]^d$ onto a discretised Hilbert space-filling curve of order p. Let $H_p : [0,1] \to [0,1]^d$ denote the Hilbert curve parameterised by arc-length (percentage along the curve). For each $\boldsymbol{u}^i$, we identify the closest point on the curve and encode the sample by its corresponding $h^i \in [0,1]$. We then apply the empirical CDF to obtain uniform intentions $z^i \in [0,1]$.

The Hilbert curve preserves locality, meaning that nearby points in the d-dimensional quantile space map to nearby positions along the one-dimensional curve (see Figure 4). As a result, similar joint quantile configurations (e.g., all actions being high, or specific correlated patterns) are assigned similar intention values.

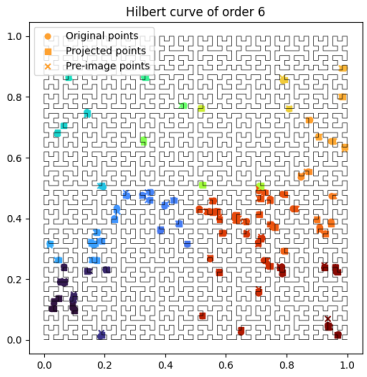

Figure 4: **The Hilbert Curve Map.** Quantiles $\boldsymbol{u}^i \in [0,1]^d$ are mapped to their closest point on the Hilbert curve.

During decentralised execution, a precomputed intention z^i is first mapped back to its arc-length parameter via the inverse ECDF. The corresponding quantiles are then obtained with an approximate inverse mapping $H_p^{-1}(h^i)$, implemented by nearest-neighbour segment matching on the discretised Hilbert curve consisting of 4^p line segments.

4.3 Part III: Coordinating Intentions

Finally, we coordinate the intentions via a generative copula model. The use of a copula model for $\boldsymbol{z} = (z^1, \ldots, z^n)$ is motivated by the need to preserve each agent's intention distribution exactly so that the quantiles $\boldsymbol{u}$ are represented accurately. Furthermore, we wish to purely model the dependence of the intentions in order to coordinate them among players.

As this is the highest-dimensional part of our model, we rely on powerful flow copula models developed to deal with many dimensions (Huk & Damoulas, 2025). Specifically, for $\boldsymbol{z} = (z^1, \ldots, z^n) \in [0,1]^n$, we define variables $\tilde{\boldsymbol{z}} = \Phi^{-1}(\boldsymbol{z})$ as the element-wise standard Gaussian distribution applied to intention observations. We then augment them via a time dimension so that $\tilde{\boldsymbol{z}}_0 = \tilde{\boldsymbol{z}}$ and $\tilde{\boldsymbol{z}}_1 = \boldsymbol{x}$ with $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}^n)$. This construction ensures that at any time $0 \leq s \leq 1$, $\tilde{\boldsymbol{z}}_s$ is uniformly marginal and so is distributed according to a copula (Huk & Damoulas, 2025). As $s \to 0$ we recover the dependence structure of the coordination between agents, while as $t \to 1$, the dependence is forgotten and agents act independently.

We learn to sample from the true copula via flow matching, training a velocity field $v_\theta(\tilde{\boldsymbol{z}}_s, s)$ to match the straight-line vector field $\dot{\tilde{\boldsymbol{z}}}_s = \boldsymbol{x} - \tilde{\boldsymbol{z}}$ between coupled data $(\tilde{\boldsymbol{z}}_0, \tilde{\boldsymbol{z}}_1)$. At inference time, we sample independent intentions $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}^n)$, then flow backwards from $s = 1$ to $s = 0$ via the ODE $d\tilde{\boldsymbol{z}}_s/ds = v_\theta(\tilde{\boldsymbol{z}}_s, s)$ to recover coordinated intentions $\tilde{\boldsymbol{z}}_0 \sim c(\Phi^{-1}(\boldsymbol{z}))$ encoding the target agent coordination structure. We recover copula samples for the coordination of intentions via $\boldsymbol{z} = \Phi(\tilde{\boldsymbol{z}})$.

5 Related Work

Multi-Agent Imitation Learning Early generative approaches include Generative Multi-Agent Behavioral Cloning (Zhan et al., 2018) and Coordinated Multi-Agent Imitation Learning Le et al. (2017), which model joint behaviours using latent-variable or hierarchical architectures. Recently Lee et al. (2025) trains a joint flow and distils it into independent policies under a Wasserstein objective.

Table 1: Test set overall and per-player metrics (mean ± std). Best (min) in bold.

Model	Overall RMSE	Overall MAE	Overall Energy Score (44D)	Per-Player Mean RMSE	Per-Player Mean MAE	Per-Player Mean Energy Score
MAGIC Hilbert	$0.4271_{\pm 0.15}$	$0.2997_{\pm 0.10}$	$2.8953_{\pm 0.33}$	$0.5845_{\pm 0.15}$	$0.4782_{\pm 0.11}$	$0.4814_{\pm 0.09}$
MAGIC ρ	$\mathbf{0.3928}_{\pm 0.14}$	$\mathbf{0.2690}_{\pm 0.09}$	$\mathbf{2.5176}_{\pm 0.32}$	$\mathbf{0.5354}_{\pm 0.15}$	$\mathbf{0.4310}_{\pm 0.10}$	$0.4359_{\pm 0.08}$
Copula	$0.3951_{\pm 0.14}$	$0.2720_{\pm 0.09}$	$2.6217_{\pm 0.30}$	$0.5387_{\pm 0.15}$	$0.4351_{\pm 0.10}$	$\mathbf{0.4341}_{\pm 0.08}$
Full flow	$5.6424_{\pm 2.31}$	$2.5868_{\pm 0.69}$	$19.0765_{\pm 30.08}$	$7.7923_{\pm 1.72}$	$4.1062_{\pm 0.53}$	$3.6326_{\pm 0.51}$
Indep flow	$0.6494_{\pm 0.20}$	$0.4638_{\pm 0.14}$	$3.2245_{\pm 0.65}$	$0.8847_{\pm 0.25}$	$0.7349_{\pm 0.19}$	$0.5958_{\pm 0.12}$
Factorised flow	$0.6453_{\pm 0.18}$	$0.4676_{\pm 0.14}$	$3.5609_{\pm 0.51}$	$0.8857_{\pm 0.22}$	$0.7418_{\pm 0.18}$	$0.6627_{\pm 0.11}$
Projected flow	$0.7316_{\pm 0.21}$	$0.5274_{\pm 0.16}$	$4.2080_{\pm 0.62}$	$1.0029_{\pm 0.25}$	$0.8330_{\pm 0.21}$	$0.7736_{\pm 0.15}$

Centralised Training with Decentralised Execution CTDE is a widely adopted paradigm in cooperative multi-agent learning where agents may use global information during training but must act independently at deployment (Amato, 2024). Most MAIL methods compatible with CTDE rely on factorised policies to ensure decentralised execution.

Copulas and Generative Models Copulas provide a principled way to decouple marginal distributions from dependence structure (Sklar, 1959; Joe, 2014). In multi-agent imitation learning, (Wang et al., 2021) propose a Gaussian copula model over all agent action dimensions, capturing joint dependence directly in the full action space. However, their approach scales with the total number of action dimensions and relies on a parametric Gaussian dependence structure. Beyond **MAIL** Tagasovska et al. (2019) places vine copulas on spaces parametrised by autoencoders. Fan & Joe (2024) uses copulas to model dependencies in the latent distribution of the variational autocoder,

Diffusion and Flow Models for Structured Dependence Diffusion and flow-based models have emerged as powerful tools for high-dimensional generative modelling. Recent work extends these methods to structured distributions and copula modelling (Huk & Damoulas, 2025), enabling expressive dependence modelling with guaranteed marginal control.

6 EXPERIMENTS

This Section assesses the following research questions with respect to MAGIC's performance:

RQ1. *Does* **MAGIC** *learn dependent policies that models in (2), (3), (4) are unable to learn?*

RQ2. *How well can* **MAGIC** *model complex multi-agent policies compared to competing approaches?*

Datasets and Benchmarks. We apply our method to simulated data (in Figure 2), and to human football data[2]. The football data consists of demonstrations from two teams with $n = 22$ players each, which we treat as agents, endowed with $d = 2$ movement dimensions treated as actions. The data represents a full game with 141156 demonstrations at a frequency of 20 frames per second, which we downsample to $T = 6994$ demonstrations at 1 frame per second to give time for players to move between steps. The state comprises the x, y coordinates of each player as well as those of the ball for a total of $s = 46$. We use an $60/20/20$ split for training, validation and testing. Our Benchmarks focus on flow-based policy models, given their recent success in modelling complex multi-agent behaviour (Lee et al., 2025) and our use of a flow for our generative coordination model. We parameterise an independent, factorised, and joint policy with a flow model, using the flow matching loss as an objective (Lipman et al., 2023). We do not use Q functions as critics, as our goal is strictly to evaluate imitation capabilities instead of maximising rewards, as in Pearce et al. (2023).

Evaluation Metrics. We evaluate each policy model based on how well it can imitate the ground truth actions. We quantify this using the root mean squared error (RMSE), mean absolute error (MAE), and energy score, which are computed once for the full 44-dimensional actions and once averaged over the 2-dimensional actions of each agent. We further report the continuous ranked probability score (CRPS) averaged across all action dimensions, and energy score Eqn. (11). To

[2]Data is obtained from, and publicly available at: https://github.com/metrica-sports/sample-data.

Table 2: Test set covariance and correlation differences. Columns: Frobenius norms (Frob Cov, Frob Corr) and mean differences with std. Best (min) in bold.

Simulation	Frob Cov	Frob Corr	Cov Diff	Corr Diff
MAGIC Hilbert	117.9	**0.4440**	$0.7201_{\pm 2.58}$	$\mathbf{-0.0003}_{\pm 0.01}$
MAGIC ρ	117.7	0.4586	$0.7083_{\pm 2.58}$	$-0.0006_{\pm 0.01}$
Copula	**117.2**	0.4540	$\mathbf{0.7073}_{\pm 2.57}$	$-0.0005_{\pm 0.01}$
Full flow	1334.1	2.7415	$8.0502_{\pm 29.23}$	$-0.0094_{\pm 0.06}$
Indep flow	204.5	0.5506	$1.4457_{\pm 4.42}$	$\mathbf{0.0003}_{\pm 0.01}$
Factorised flow	169.3	0.5124	$1.2591_{\pm 3.64}$	$0.0006_{\pm 0.01}$
Projected flow	195.3	0.5974	$1.7026_{\pm 4.0998}$	$0.0023_{\pm 0.0134}$

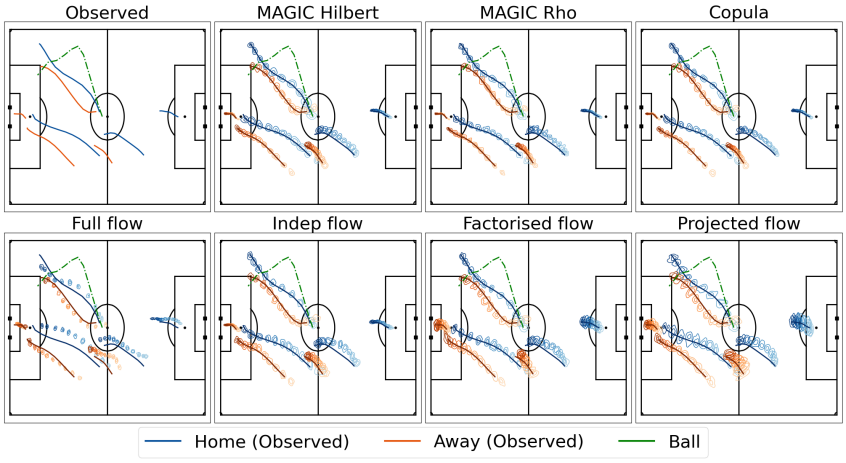

Figure 5: Predicted agent trajectories of **MAGIC** and across our baselines. **MAGIC** is able to better predict the ground truth with more confidence.

quantify dependence structure, we compute the Frobenius norm of the errors in covariance (Frob cov) and correlation matrices (Frob Corr) between policy samples and actions.

Answer to RQ1 *(Policy flexibility)*. In Figure 2, **MAGIC** is shown as the only policy capable of modelling the ground truth action distributions on two examples. In the top panel, the independent policies fail to coordinate between agents, resulting in actions wrongly placed in the lower left quadrant, whereas **MAGIC** successfully coordinates with the intention model. These results confirm our Corollary 3.2, exemplifying the necessity of capturing dependence for joint policy learning.

Answer to RQ2 *(Imitation performance)*. In Table 1, we show that **MAGIC** ρ outperforms all baselines except for the mean energy score, where it comes second to the copula policy. This shows the benefit and robustness of aggregating dependence into intentions, as **MAGIC** outperforms the full flow and copula policies that model all actions at once. To further assess the impact of intention modelling, in Table 2, we show that dependence is preserved by modelling intentions as **MAGIC** is competitive to the copula policy which explicitly fits the whole dependence structure. Finally, in Figure 5, **MAGIC** shows accurate predictions with calibrated uncertainty quantification, whereas the projected flow overshoots the uncertainty, while the full flow underestimates uncertainty.

7 Conclusion

We introduced **MAGIC**, a scalable **MAIL** framework that learns coordinated multi-agent behaviour without modelling the combinatorial joint action space. **MAGIC** captures inter-agent dependencies while retaining decentralised execution required for CTDE by compressing each agent's action distribution into a one-dimensional intention and learning a diffusion-copula model over these intentions. We provide theoretical results characterising the expressivity of the induced dependence and show strong empirical gains over baselines.

REFERENCES

Christopher Amato. An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2409.03052*, 2024.

Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Nicolas Chapados, and Alexandre Drouin. TACTis-2: Better, faster, simpler attentional copulas for multivariate time series. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xtOydkE1Ku.

Xinyao Fan and Harry Joe. High-dimensional factor copula models with estimation of latent variables. *Journal of Multivariate Analysis*, 201:105263, 2024. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2023.105263. Copula Modeling from Abe Sklar to the present day.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Marius Hofert, Ivan Kojadinovic, Martin Mächler, and Jun Yan. *Elements of copula modeling with R*. Springer, 2018.

David Huk and Theodoros Damoulas. Diffusion and flow-based copulas: Forgetting and remembering dependencies. *arXiv preprint arXiv:2509.19707*, 2025.

Harry Joe. *Dependence modeling with copulas*. CRC press, 2014.

Hoang M Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*, pp. 1995–2003. PMLR, 2017.

Dongsu Lee, Daehee Lee, and Amy Zhang. Multi-agent coordination via flow matching. *arXiv preprint arXiv:2511.05005*, 2025.

Chao Li, Ziwei Deng, Chenxing Lin, Wenqi Chen, Yongquan Fu, Weiquan Liu, Chenglu Wen, Cheng Wang, and Siqi Shen. Dof: A diffusion factorization framework for offline multi-agent reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, 2020. ISBN 9781713829546.

Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

M Sklar. Fonctions de répartition à n dimensions et leurs marges. In *Annales de l'ISUP*, volume 8, pp. 229–231, 1959.

Natasa Tagasovska, Damien Ackerer, and Thibault Vatter. Copulas as high-dimensional generative models: Vine copula autoencoders. *Advances in neural information processing systems*, 32, 2019.

Hongwei Wang, Lantao Yu, Zhangjie Cao, and Stefano Ermon. Multi-agent imitation learning with copulas. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 139–156. Springer, 2021.

Eric Zhan, Stephan Zheng, Yisong Yue, and Patrick Lucey. Generative multi-agent behavioral cloning. *arXiv preprint arXiv:1803.07612*, 2, 2018.

A METRICS

Energy Score The Energy score (ES) ((Liu et al., 2020)) is defined as

$$\text{ES}(P, \boldsymbol{x}) = \mathbb{E}_{\boldsymbol{X} \sim P} [\, || \boldsymbol{X} - \boldsymbol{x} \, || \,] - \frac{1}{2} \mathbb{E}_{\boldsymbol{X}, \boldsymbol{X}' \sim P} [\, || \boldsymbol{X} - \boldsymbol{X}' \, ||. \,] \tag{11}$$

Frobenius Norm The Frobenius Norm is defined by

$$|| \, A \, ||_F = \sqrt{\text{trace}(A^* A)}. \tag{12}$$

We obtain Frob Corr and Frob Cov by computing the element-wise error between the true correlation and covariance matrices and then computing the Frobenius norm of this error matrix.

B MODEL ARCHITECTURE

In experiments, we implement the independent marginal conditional densities for a movement axis of a single player as Gaussian distributions parameterised with a mean and variance obtained as a function of covariates. The covariates we use are the last x_p, y_p coordinates of the given player as well as the last x_b, y_b coordinates of the ball. We model the mean and variance parameters via a multi-layered perceptron taking as inputs the 4D vector of x_p, y_p, x_b, y_b, composed of two hidden layers with 4 nodes each, and outputting the mean and variance for the predictive density. Training for a single marginal with the CRPS finishes within a minute on a standard CPU.

The copula model is implemented as a flow copula following Huk & Damoulas (2025) as a multi-layered perceptron with width 512 and depth 5. Training takes 20 minutes on an Apple Silicon Mac, using the MPS GPU backend.