WORLD-AS-GRAPH: RELATIONAL WORLD MODELING THROUGH LATENT SPACE GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models aim to learn representations of real-world environments and predict their future evolution. Recent object-centric world models have made expressive progress by representing visual scenes as sets of object-level latent states, but object-object relations are often captured only implicitly, which limits *explicit relational and temporal structure modeling* and *object-centric dynamic memory modeling*. To address such challenges, we propose **W**orld-**a**s-**G**raph (**WAG**), a graph-based object-centric world model that introduces relational inductive bias into JEPA-style predictive representation learning. The proposed WAG contains two main modules: (1) Relation-aware structure induction, which constructs time-varying latent graphs from object-centric slots and designs relation-aware object masking policies to guide relational object representation learning in latent space; (2) Object-centric memory transition, which maintains and updates object-level dynamic states by combining relational information from neighboring objects with historical memory, enabling effective autoregressive future prediction. Extensive experiments on both visual reasoning and robotic manipulation tasks demonstrate the superior performance of our proposed WAG.

1 INTRODUCTION

World models (Ha & Schmidhuber, 2018; Ding et al., 2025; LeCun et al., 2022) have recently attracted increasing attention for capturing and modeling real-world environments and predicting their future evolutions. By learning future states from past observations, world models provide better world understanding, prediction, and decision-making across a wide range of applications, from video generation to robotics (Puspitasari et al., 2024; Bruce et al., 2024; Wu et al., 2024; Wang et al., 2026b; Yang et al., 2026; Zhang et al., 2026; Bi et al., 2026). In this context, a key aspect of world modeling is to learn predictive and structural representations that capture world dynamics in a **latent space** by explicitly modeling how individual objects interact and evolve over time (Yang et al., 2026; Lei et al., 2026; Feng et al., 2025a; Kipf et al., 2020).

Inspired by this perspective, the joint-embedding predictive architecture, i.e., JEPA (LeCun et al., 2022; Assran et al., 2023; Klindt et al., 2026; Bai & Xiong, 2026), has demonstrated impressive progress for predictive representation learning in world models (Saito et al., 2025; Yang et al., 2026). Instead of reconstructing observations at the pixel level, JEPA predicts future representations directly in latent space, supporting effective representation learning for perception and action. Meanwhile, to better capture object interactions in the world, object-centric models have recently been adopted as a foundation for world modeling (Nam et al., 2026). These models typically represent a visual scene as a set of object-level latent states (i.e., slots), allowing different objects to be modeled separately while capturing visual dynamics and learning object-centric predictive representations (Mosbach et al., 2024; Nishimoto & Matsubara, 2026; Kipf et al., 2021; Ferraro et al., 2025).

Despite the promising performance of object-centric representation learning with JEPA, recent studies suggest that object-centric predictors may not guarantee that meaningful interactions are explicitly captured from object slots alone (Lei et al., 2026). Existing object-centric world models, including Causal-JEPA (Nam et al., 2026), therefore introduce different mechanisms to encourage interaction learning, such as structured dynamics, sparse attention, and task-specific supervision (Kipf et al., 2020; Feng et al., 2025a; Mosbach et al., 2024; Ferraro et al., 2025). However, without explicit relational inductive biases, models may still predict future states primarily from ob-

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

ject self-dynamics, leaving two essential challenges: **C1: Limited relational and temporal structure modeling**, where existing object-centric latent world models operate on latent object slots, but often leave object-object relations implicit through attention or object-level masking. In particular, they do not distinguish objects according to relational roles or explicitly model how object relations evolve over time. As a result, masked latent prediction in JEPA may provide limited self-supervised signals for effectively learning object interactions. **C2: Limited object-centric dynamic memory modeling**, where future prediction requires each object to carry its own interaction history together with relational information accumulated from other objects. Without explicit object-level state memory modeling, the model may lose important historical cues about how objects move, interact, and influence future states, leading to suboptimal long-horizon prediction.

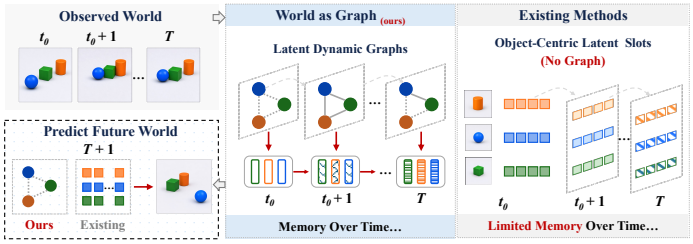


Figure 1: Comparison of our proposed World-as-Graph (WAG) with existing object-centric world models (with no explicit graph structures (C1) and limited temporal memory (C2)).

To address these challenges, we propose a **W**orld-**A**s-**G**raph model, dubbed **WAG**, for relational world modeling through graph-based representation learning in the latent space, as shown in Figure 1. The key idea is to exploit **graph structure** as a relational inductive bias (RIB) (Battaglia et al., 2018) for object-centric JEPA-style world models, enabling structured state representations of objects and their interactions. Specifically, our proposed WAG contains two core sub-modules: (1) *Relation-aware structure induction*, which induces relational structures in latent space through a (1-a) latent dynamic graph constructor and a (1-b) relation-aware object masker; and (2) *Object-centric memory transition*, which models the temporal evolution of object-level dynamic states through a (2-a) temporal graph neural network (GNN) transition encoder and a (2-b) memory-driven future predictor. Concretely, the latent dynamic graph constructor treats object slots as nodes and builds time-varying relational graphs, followed by the relation-aware object masker, which selects objects based on relational centrality or temporal neighborhood changes. Masked object states are recovered from relational context and historical information. The first module models how object relations and structural dependencies evolve over time and guides JEPA-style predictive learning, addressing challenge **C1**. Afterward, the temporal GNN transition encoder updates each object state using neighboring relations and its historical memory. The memory-driven future predictor recursively evolves these memory states and constructs future dynamic graphs to predict future object states. The second module models how object-centric dynamic states evolve and supports future world prediction, addressing challenge **C2**. Extensive experiments on visual reasoning and robotic manipulation tasks show that our proposed WAG improves performance by up to 30.03 percentage points, while achieving $2.4\times$ faster training and up to $16.4\times$ lower peak GPU memory. In summary, the contributions of this work are listed below:

- To the best of our knowledge, we are the first to introduce latent space graphs for relational modeling in object-centric JEPA-style world models to explicitly capture object interactions and temporal dynamics in a **W**orld-**A**s-**G**raph (**WAG**).
- The proposed WAG consists of two core sub-modules that jointly conduct (1) *relation-aware structure induction* composed of a latent dynamic graph constructor and a relation-aware object masker, and (2) *object-centric memory transition* composed of a temporal GNN transition encoder and a memory-driven future predictor, for expressive future world prediction.
- Extensive experiments demonstrate the effectiveness and benefits of WAG in introducing relational inductive bias to model object interactions and future dynamics.

Prior Work. Our work is closely related to three lines of research: (a) *joint-embedding predictive architectures (JEPA)*, (b) *graph world models*, and, to a lesser extent, (c) *dynamic graph representation learning*. Specifically, *JEPA methods* (LeCun et al., 2022; Assran et al., 2023; Klindt et al., 2026; Bai & Xiong, 2026; Saito et al., 2025; Tuncay et al., 2025; Balestriero & LeCun, 2025; Yang et al., 2026; Nam et al., 2026) learn predictive representations directly in latent space, while recent object-centric JEPA-style world models (Feng et al., 2025a; Wang et al., 2025) further represent visual scenes as sets of object-level latent states for learning object-wise dynamics. However, these existing methods often capture

implicit interactions, while our WAG explicitly constructs evolving latent dynamic graphs for object state interactions. Most existing *graph world models* (Liu et al., 2026; Feng et al., 2025b; Song & Cai, 2026; Wang et al., 2026a) introduce world-modeling principles into graph-structured data, i.e., *world modeling for graphs*, while our work first introduces latent graph structures as relational inductive biases for object-centric world modeling, i.e., *graphs for world modeling*. *Dynamic graph representation learning* (Feng et al., 2025c; Zheng et al., 2025) is closely related to the temporal GNN component in WAG. More detailed discussions of these related research directions are provided in Appendix A.1.

2 METHODOLOGY

2.1 PRELIMINARY

Notation. Given a video representing a dynamic world as an image sequence, a pretrained object-centric encoder f_{Φ^*} extracts N latent slot representations from each frame at time step t as $\mathbf{S}_t = [\mathbf{s}_t^1, \ldots, \mathbf{s}_t^N]^\top \in \mathbb{R}^{N \times D_0}$, where $\mathbf{s}_t^i \in \mathbb{R}^{D_0}$ denotes the latent feature of the i-th slot, D_0 is the feature dimension, and N is the fixed number of object slots. In this work, we transfer the latent slot representations to a discrete-time dynamic graph sequence $\mathcal{G} = \{G_t\}_{t \in \mathcal{T}}$. Each graph snapshot at time step t is defined as $G_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{S}_t)$, where $\mathcal{V}_t = \{v_t^1, \ldots, v_t^N\}$ denotes the set of object nodes in the latent space, and $\mathcal{E}_t \subseteq \mathcal{V}_t \times \mathcal{V}_t$ denotes the set of edges. Each node v_t^i corresponds to the i-th object-centric slot at time step t and is associated with the latent feature $\mathbf{s}_t^i$. For future prediction, we have $\mathcal{T} = \mathcal{T}_{\text{hist}} \cup \mathcal{T}_{\text{pred}}$, where T_h and T_p indicate the history window length and future prediction horizon, respectively. Given the current time step t, the corresponding history and prediction time sets are defined as $\mathcal{T}_{\text{hist}} = \{t - T_h + 1, \ldots, t\}$ and $\mathcal{T}_{\text{pred}} = \{t+1, \ldots, t+T_p\}$. Given the observed slot states $\{\mathbf{S}_t\}_{t \in \mathcal{T}_{\text{hist}}}$, the world model aims to predict the future slot states $\{\mathbf{S}_t\}_{t \in \mathcal{T}_{\text{pred}}}$.

Problem Setting. Following the world model framework of Ha & Schmidhuber (2018), a model learns a compact representation of the environment from observations and predicts how that representation evolves in response to current actions. In this work, we follow this general formulation together with the object-centric and JEPA-style representation learning paradigm (Nam et al., 2026; Assran et al., 2023), and introduce graph machine learning and relational inductive bias to explicitly model interactions within the latent world states. Based on this, we further define a graph-structured world model, termed the **W**orld-**A**s-**G**raph (**WAG**) model, as follows.

DEFINITION 1 (WORLD-AS-GRAPH MODEL). Given the historical object-centric latent states $\{\mathbf{S}_t\}_{t \in \mathcal{T}_{\text{hist}}}$, a World-As-Graph (WAG) model is defined as $\mathcal{M}_{\text{WAG}} = \langle \mathcal{R}, \mathcal{F}_{\theta,\omega} \rangle$, where:

- *Relation-Aware Structure Induction* $\mathcal{R} = \langle \Gamma_\psi, \pi_{\text{mask}} \rangle$: first constructs dynamic relational graphs from object-centric latent representations through the latent dynamic graph constructor Γ_ψ, and then uses the relation-aware masking policy π_{mask} to identify relationally informative object nodes based on the graph structure.
- *Object-Centric Memory Transition* $\mathcal{F}_{\theta,\omega} = \langle f_{\text{enc}}^\theta, f_{\text{pred}}^\omega \rangle$: maintains and updates object-level memory states from the masked historical graph sequence through the temporal GNN transition encoder f_{enc}^θ, and further evolves these memory states through the memory-driven future predictor f_{pred}^ω to predict the future object-centric latent states $\{\widehat{\mathbf{S}}_t\}_{t \in \mathcal{T}_{\text{pred}}}$.

2.2 WORLD-AS-GRAPH FRAMEWORK

To explicitly model structured state representations of objects and their interactions, in this work, we first propose WAG, a graph-based object-centric world model. The overall framework is presented in Fig. 2. Given an input video, a frozen object-centric encoder first extracts object-level latent representations over time. The proposed WAG then performs two main stages. First, *relation-aware structure induction* constructs latent dynamic graphs among object slots and applies relation-aware masking based on relational centrality or temporal dynamics. Second, *object-centric memory transition* propagates relational information through a temporal GNN and updates object-level memory states over the observed history. These memory states are used for masked-state prediction during training and are further evolved by the memory-driven future predictor to autoregressively generate future object representations in latent space.

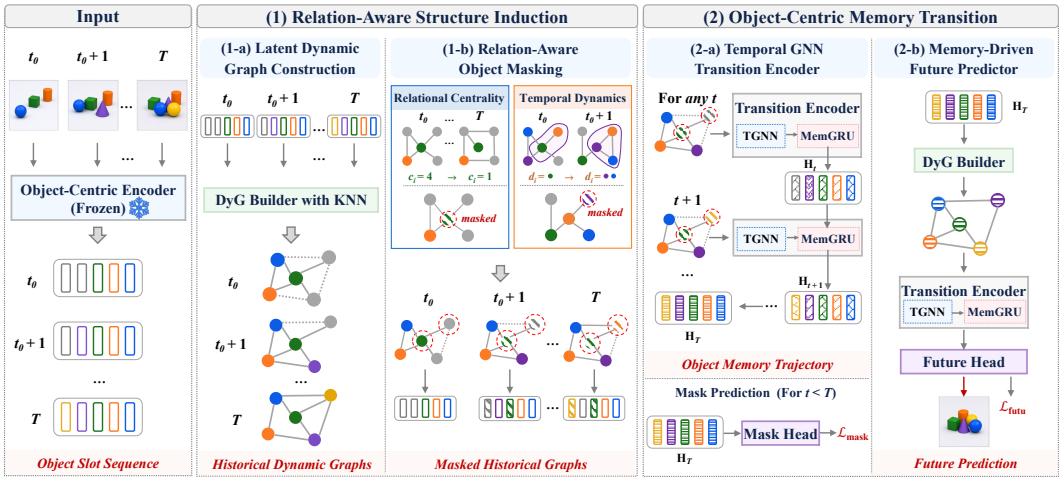


Figure 2: Overall framework of our proposed WAG.

2.2.1 Relation-Aware Structure Induction

This module explicitly induces relational bias through dynamic graph structure learning among object-centric slots and further guarantees relation-aware representation learning. Specifically, it contains two core sub-components: (a) *Latent dynamic graph construction*, which treats each object slot as a node and builds temporal object-object edges based on latent representation similarity; (b) *Relation-aware object masking*, which leverages the relational structure to inform the model of which objects deserve more attention.

Latent Dynamic Graph Construction. To explicitly model relational structures in latent space, given the object slot representations $\mathbf{S}_t = [\mathbf{s}_t^1, \ldots, \mathbf{s}_t^N]^\top$, each slot $\mathbf{s}_t^i$ is treated as an object node v_t^i. Our proposed WAG constructs a latent dynamic graph through a parameter-free dynamic graph builder Γ based on a similarity function $\mathrm{sim}(\cdot,\cdot)$ and K-nearest neighbor (KNN) algorithm as:

$$G_t = \Gamma_K^{\mathrm{sim}}(\mathbf{S}_t) = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{S}_t), \quad \mathcal{E}_t = \left\{ (i,j) \mid j \in \underset{k \neq i}{\arg\,\text{Top-}K}\ \mathrm{sim}\left(\mathbf{s}_i^t, \mathbf{s}_k^t\right) \right\}, \tag{1}$$

where $\mathcal{E}_t$ denotes the set of edges connecting each object to its Top-K most similar nodes. In this work, we use cosine similarity, and the edge weight is given by $w_{ij}^t = \cos\left(\mathbf{s}_i^t, \mathbf{s}_j^t\right)$. This produces a sequence of time-varying latent graphs as $\mathcal{G} = \{G_t\}_{t \in \mathcal{T}}$.

Relation-Aware Object Masking. For each latent dynamic graph G_t, object nodes are selected for masking according to their relational roles in the constructed graph. In this way, the proposed WAG is able to recover the masked object states by jointly using relational information from other objects and the object's own historical states. Specifically, let $\mathcal{I}_{\mathrm{mask}} \subset \{1, \ldots, N\}$ denote the selected masked node indices, with $|\mathcal{I}_{\mathrm{mask}}| = M$ and $M < N$. Unlike random slot masking, our relation-aware object masking strategy is guided by the latent dynamic graph topology over the history window. Specifically, we design two strategies as below.

(I) Relational Centrality Masking. We first select relationally central object nodes according to their roles in the constructed latent dynamic graphs. Since each node in the KNN graph has a fixed out-degree K, we use the in-degree as the centrality signal to measure how frequently an object node v_t^i appears in the neighbor sets $\mathcal{N}_t^j$ of other objects over the observed history $\mathcal{T}_{\mathrm{hist}}$. Therefore, we define the relational centrality score c_i and select the masked node indices $\mathcal{I}_{\mathrm{mask}}^{\mathrm{cen}}$ as

$$c_i = \sum_{t \in \mathcal{T}_{\mathrm{hist}}} \sum_{\substack{j=1 \\ j \neq i}}^{N} \mathbb{I}\left[i \in \mathcal{N}_t^j \right], \qquad \mathcal{I}_{\mathrm{mask}}^{\mathrm{cen}} = \underset{i \in \{1,\ldots,N\}}{\arg\,\text{Top-}M}\ c_i, \tag{2}$$

where the Top-M objects with the highest centrality scores are selected for masking. By recovering these relationally central object states in JEPA-style learning, the proposed WAG exploits relational information from other objects together, rather than relying mainly on object self-dynamics.

(II) Temporal Dynamics Masking. We select object nodes according to the temporal changes of their relational contexts. The intuition behind is straightforward: if an object's neighborhood changes frequently across consecutive time steps, its state is more likely to be affected by evolving interactions with other objects. Therefore, we define the temporal dynamics score d_i by measuring the accumulated changes in the neighbor set $\mathcal{N}_t^i$ over the observed history $\mathcal{T}_{\text{hist}}$, and select the masked node indices $\mathcal{I}_{\text{mask}}^{\text{dyn}}$ as

$$d_i = \sum_{t \in \mathcal{T}_{\text{hist}}} \left(K - \left| \mathcal{N}_t^i \cap \mathcal{N}_{t-1}^i \right| \right), \qquad \mathcal{I}_{\text{mask}}^{\text{dyn}} = \underset{i \in \{1,\dots,N\}}{\arg \text{Top-}M}\ d_i, \tag{3}$$

where $K - \left| \mathcal{N}_t^i \cap \mathcal{N}_{t-1}^i \right|$ measures the number of changed neighbors between two consecutive time steps. By recovering these temporally dynamic object states, the proposed WAG exploits evolving relational information across time.

After obtaining $\mathcal{I}_{\text{mask}} \in \{\mathcal{I}_{\text{mask}}^{\text{cen}}, \mathcal{I}_{\text{mask}}^{\text{dyn}}\}$, we replace the corresponding object features with a learnable mask token over the observed history, while keeping the first historical graph G_{t_0} at $t_0 = t - T_h + 1$ fully visible to initialize the object states. This produces a partially observed historical graph sequence $\mathcal{G}^\dagger = \{G_t^\dagger\}_{t \in \mathcal{T}_{\text{hist}} \setminus \{t_0\}}$, where we have $G_t^\dagger = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{S}_t^\dagger)$ with masked object representations $\mathbf{S}_t^\dagger$. One goal of the proposed WAG is to recover the masked object states from their relational context and historical information.

2.2.2 Object-Centric Memory Transition

Although the relation-aware structure induction explicitly models object-object relations *within* each time step, future world prediction further requires each object to maintain its own historical state across time. This becomes critical in two cases. First, a masked object node cannot observe its own state in the remaining historical frames. Second, during future generation through autoregressive rollout, no future observations are available for any object node. In light of this, we design an object-centric memory transition module, where each object node maintains an individual *memory state* to preserve its historical information and relational interactions over time. Specifically, this module contains two sub-components: (a) *Temporal GNN transition encoder*, which updates object-level memory states over the observed history through graph structure-guided message passing, and (b) *Memory-driven future predictor*, which evolves the learned memory states to predict future object states without new observations.

Temporal GNN Transition Encoder. Given the partially observed historical graph $G_t^\dagger$ after relation-aware object masking, we maintain each object node v_t^i an individual memory state $\mathbf{h}_t^i \in \mathbb{R}^{D_1}$, where D_1 is the memory feature dimension. Therefore, we define the object-level memory states as $\mathbf{H}_t = [\mathbf{h}_t^1, \dots, \mathbf{h}_t^N]^\top \in \mathbb{R}^{N \times D_1}$. Since the first historical graph at t_0 is kept fully visible, the object memory states are initialized as the observed object representations $\mathbf{H}_{t_0} = \mathbf{S}_{t_0}$.

For the remaining historical time steps, we design a temporal GNN transition encoder f_{enc}^θ that updates the object-level memory states by jointly considering the current partially observed graph $G_t^\dagger$ and the previously accumulated memory states $\mathbf{H}_{t-1}$ through temporal graph message passing:

$$\mathbf{H}_t = f_{\text{enc}}^\theta \left(G_t^\dagger, \mathbf{H}_{t-1} \right) = \text{MemGRU}_{\theta_m} \left(\text{TGNN}_{\theta_g} \left(G_t^\dagger \right), \mathbf{H}_{t-1} \right), \qquad t \in \mathcal{T}_{\text{hist}} \setminus \{t_0\}, \tag{4}$$

where $\theta = \{\theta_g, \theta_m\}$ denotes the parameters of the temporal GNN and memory-based gated recurrent unit (GRU). In this way, each object memory state $\mathbf{h}_t^i$ integrates the current object information, relational context from neighboring objects, and its previous memory state $\mathbf{h}_{t-1}^i$, therefore preserving both object-specific historical information and relational information accumulated over time.

To further encourage our proposed model to learn predictive object representations through the masking policy, we apply an auxiliary masked-state prediction head g_{mask}^ϵ to the corresponding memory states for generating predicted masked object representations as:

$$\widehat{\mathbf{S}}_t^{\text{mask}} = g_{\text{mask}}^\epsilon \left(\mathbf{H}_t, \mathcal{I}_{\text{mask}} \right), \qquad t \in \mathcal{T}_{\text{hist}} \setminus \{t_0\}. \tag{5}$$

Memory-Driven Future Predictor. After the observed history ends, we design a memory-driven future predictor f_{pred}^ω to autoregressively evolve the learned object-level memory states using the

previously learned memory for relational future prediction. Specifically, for each future time step $t \in \mathcal{T}_{\text{pred}}$, we first apply the same latent dynamic graph constructor to the previous memory states and the same temporal GNN encoder to capture the object-level interactions. Then, the designed memory-driven future predictor f^{ω}_{pred} evolves the previous memory states with temporal information to update the future memory states, so we could generate the future object representation with transitioned memory as:

$$\widehat{G}_{t-1} = \Gamma^{\text{sim}}_{K}\left(\mathbf{H}_{t-1}\right), \qquad \mathbf{H}_t = f^{\omega}_{\text{pred}}\left(\widehat{G}_{t-1}, \mathbf{H}_{t-1}\right), \qquad \widehat{\mathbf{S}}^{\text{futu}}_t = g^{\eta}_{\text{proj}}\left(\mathbf{H}_t\right), \qquad t \in \mathcal{T}_{\text{pred}}, \tag{6}$$

where g^{η}_{proj} is a projection head for generating future object states from the updated memory states. In this way, the proposed WAG progressively constructs future relational structures and evolves object-level memory, predicting the future object-centric latent states $\{\widehat{\mathbf{S}}^{\text{futu}}_t\}_{t\in\mathcal{T}_{\text{pred}}}$.

2.2.3 Learning Objective

Training Stage. The proposed WAG is trained with two predictive objectives: (1) Historical masked-state prediction loss $\mathcal{L}_{\text{mask}}$, to encourage WAG to recover the original latent representations of the masked objects from their learned memory states; (2) Future-state prediction loss $\mathcal{L}_{\text{futu}}$, to predict the future object-centric latent states through autoregressive memory transition with latent dynamic graph guidance. Given the predicted masked representations $\widehat{\mathbf{S}}^{\text{mask}}_t$ and the predicted future states $\widehat{\mathbf{S}}^{\text{futu}}_t$, we define the following training objectives:

$$\mathcal{L}_{\text{mask}} = \frac{1}{|\mathcal{T}_{\text{hist}} \setminus \{t_0\}|} \sum_{t\in\mathcal{T}_{\text{hist}}\setminus\{t_0\}} \ell\left(\widehat{\mathbf{S}}^{\text{mask}}_t, \mathbf{S}^{\text{mask}}_t\right), \quad \mathcal{L}_{\text{futu}} = \frac{1}{|\mathcal{T}_{\text{pred}}|} \sum_{t\in\mathcal{T}_{\text{pred}}} \ell\left(\widehat{\mathbf{S}}^{\text{futu}}_t, \mathbf{S}^{\text{futu}}_t\right), \tag{7}$$

where $\mathbf{S}^{\text{mask}}_t$ and $\mathbf{S}^{\text{futu}}_t$ indicate the original latent masked representations and the target future representations, respectively. $\ell(\cdot,\cdot)$ denotes the latent space prediction loss, for which we use mean squared error (MSE). Therefore, the overall training objective is formulated as

$$\mathcal{L} = \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{futu}}\mathcal{L}_{\text{futu}}, \tag{8}$$

where λ_{mask} and λ_{futu} are hyper-parameters. We provide a theoretical analysis of stability and error accumulation in Appendix A.2. The overall training algorithm is in Appendix A.3.

Inference Stage. During inference, the relation-aware object masking and auxiliary masked-state prediction head are NOT used. Given the FULLY observed historical object representations, we apply Eq. (4) to learn the object-level memory states over the history window, and then, starting from the last historical memory state. The memory-driven future predictor performs autoregressive rollout by Eq. (6). Following the standard objective of latent world modeling, this procedure will be recursively conducted for the next prediction step until the target prediction horizon is reached.

3 Experiments

To verify the effectiveness of our proposed WAG, we conduct extensive experiments on visual reasoning and robotic manipulation tasks and aim to answer the following research questions: **RQ1** [Sec 3.2]: How does the overall performance of WAG in predicting future object dynamics compare with existing predictive world-modeling baselines? **RQ2** [Sec 3.3]: How do ablation studies reveal the contributions of key components in WAG, particularly latent dynamic graph construction and different object masking strategies? **RQ3** [Sec 3.4]: How effectively can WAG model temporal physical interactions? **RQ4** [Sec 3.4]: How robust and computationally efficient is WAG in terms of hyperparameter sensitivity and running cost? More experimental results, analysis, and implementation details are listed in Appendix A.4.

3.1 Experimental Setup

Tasks and Datasets. We evaluate the proposed WAG over two tasks, i.e., *visual reasoning* and *robotic manipulation*. For the *visual reasoning* task, we use the CLEVRER (Yi et al., 2019) dataset, a synthetic video benchmark of multiple colliding objects for physical and causal understanding in

Table 1: VQA accuracy (%) comparison on CLEVRER using VideoSAUR and SAVi encoders. WAG$_{\mathrm{I}}$ and WAG$_{\mathrm{B}}$ denote instance-wise masking and batch-shared masking, respectively. Δ_{I} and Δ_{B} indicate the absolute improvements over C-JEPA in percentage points (pp). The best results are shown in bold, and the second-best results are underlined.

Model	Average	Counterfactual (%)		Explanatory (%)		Predictive (%)		Descriptive (%)
	per que. (%)	per opt.	per que.	per opt.	per que.	per opt.	per que.	
VideoSAUR Encoder								
OC-JEPA	82.79	79.53	47.68	92.88	80.58	86.15	75.04	89.59
C-JEPA	89.40	88.67	68.81	96.62	90.74	<u>93.03</u>	<u>86.93</u>	92.84
WAG$_{\mathrm{I}}$ (ours)	<u>90.79</u>	<u>90.06</u>	<u>72.22</u>	<u>97.79</u>	<u>93.96</u>	92.92	86.81	<u>93.71</u>
Δ_{I} Improv.↑	+1.39	+1.39	+3.41	+1.17	+3.22	-0.11	-0.12	+0.87
WAG$_{\mathrm{B}}$ (ours)	**91.37**	**90.55**	**73.76**	**97.84**	**94.13**	**94.43**	**89.54**	**94.05**
Δ_{B} Improv.↑	+1.97	+1.88	+4.95	+1.22	+3.39	+1.40	+2.61	+1.21
SAVi Encoder								
OC-JEPA	77.28	76.69	41.10	91.20	76.04	83.48	70.51	84.05
C-JEPA	83.88	85.16	60.19	95.34	87.27	87.46	77.25	87.81
WAG$_{\mathrm{I}}$ (ours)	<u>92.34</u>	<u>90.98</u>	<u>74.62</u>	<u>98.25</u>	<u>95.18</u>	<u>94.63</u>	<u>90.05</u>	<u>95.06</u>
Δ_{I} Improv.↑	+8.46	+5.82	+14.43	+2.91	+7.91	+7.17	+12.80	+7.25
WAG$_{\mathrm{B}}$ (ours)	**92.72**	**91.19**	**75.45**	**98.35**	**95.35**	**95.84**	**91.99**	**95.29**
Δ_{B} Improv.↑	+8.84	+6.03	+15.26	+3.01	+8.08	+8.38	+14.74	+7.48

dynamic scenes, where four types of questions (i.e., counterfactual, explanatory, predictive, and descriptive) are included for the visual question-answering (VQA) task. Each video is encoded into object-centric latent states using a frozen pretrained encoder. Following Nam et al. (2026), we use VideoSAUR (Zadaianchuk et al., 2023) and SAVi (Kipf et al., 2021) with seven object slots. Our WAG is trained on these latent states and autoregressively rolls out future object trajectories. Following SlotFormer (Wu et al., 2022), we use ALOE (Ding et al., 2021) to reason over predicted object trajectories and report per-question-type and average VQA accuracy. For the *robotic manipulation* task, we use the contact-rich PushT robotic manipulation benchmark (Chi et al., 2025), where an agent pushes a T-shaped block from a random pose to a target pose through sequential contact interactions. Following (Zhou et al., 2024; Nam et al., 2026), we evaluate PushT planning by success rate using six latent tokens (four object slots and two auxiliary tokens). An episode is successful when the final block state falls within a predefined threshold of the target.

Baselines. To ensure a fair comparison under consistent experimental settings, we mainly compare the proposed WAG with existing state-of-the-art object-centric world models that adopt comparable latent representations and evaluation protocols. For VQA evaluation, all compared methods use the same pretrained encoder, rollout procedure, and downstream models to ensure a controlled and fair comparison. For the *visual reasoning* task, we consider two state-of-the-art baseline methods: (1) OC-JEPA (Nam et al., 2026), and (2) Causal-JEPA (C-JEPA) (Nam et al., 2026). For the *robotic manipulation* task, we take the following baseline models: (1) DINO-WM (Zhou et al., 2024), (2) DINO-WM-Reg. (Darcet et al., 2024), (3) OC-DINO-WM (Nam et al., 2026), (4) OC-JEPA, and (5) C-JEPA. All baselines start from DINOv2 embeddings (Oquab et al., 2023) and differ only in predictor training. More details of all baseline methods are listed in Appendix A.4.

3.2 Performance of WAG on Visual Reasoning and Robotic Manipulation

To answer **RQ1** on the *visual reasoning* task, we report the VQA accuracy (%) across four question types on the CLEVRER dataset in Table 1. Overall, both variants of our proposed WAG consistently achieve strong VQA performance across the VideoSAUR and SAVi encoders. In particular, WAG$_{\mathrm{B}}$ improves the average accuracy over C-JEPA by 1.97 and 8.84 pp with VideoSAUR and SAVi encoders, respectively. Concretely, we can make the following essential observations. *First*, our proposed WAG consistently improves over C-JEPA across both object-centric encoders, showing that its effectiveness does not depend on a specific pretrained encoder. *Second*, our proposed WAG achieves the largest gains on counterfactual and predictive reasoning tasks, improving the per-question accuracy over C-JEPA by 15.26 and 14.74 pp, respectively, with the SAVi encoder. We attribute this to the proposed latent dynamic graphs and object-centric memory transition module, which explicitly models the temporal evolution of object states and relations, enabling stronger reasoning about unobserved and future dynamics. *Third*, batch-shared masking consistently outperforms instance-wise masking across both encoders, indicating more stable learning for future prediction.

To answer **RQ1** on the *robotic manipulation* task, we compare planning performance on PushT under different world-model token budgets in Table 2. The results show that WAG achieves the best success rate of 90.70% using only 6×128 latent tokens, which is close to DINO-WM, achieving 91.33% with a much larger token budget of 196×384. Under the same compact 6×128 setting, WAG clearly outperforms all object-centric baselines. It exceeds JEPA-style baselines, OC-JEPA and C-JEPA, by 14.70 and 2.03 percentage points, respectively. This further demonstrates that our proposed relational structure modeling in WAG can improve compact object-centric representations for downstream robotic planning.

Table 2: PushT planning success rates across different world-model token budgets.

# Token $\times d$	Model	Success Rate (%)
196×384	DINO-WM	**91.33**
	DINO-WM-Reg.	88.00
6×128	OC-DINO-WM (ref.)	60.67
	OC-JEPA	76.00 ↑+15.33
	C-JEPA	88.67 ↑+28.00
	WAG (ours)	<u>90.70</u> ↑+30.03

3.3 Ablation Study of WAG on Key Components

Overall Ablation. To answer **RQ2**, we report the overall ablation results in Table 3. It shows that all components contribute to the final performance of WAG. Full WAG achieves the best average accuracy of 91.37%, improving C-JEPA by 1.97 percentage points. Removing the memory predictor causes the largest degradation, particularly on predictive questions, where accuracy drops from 89.54% to 85.55%. Removing DyGraph also leads to a clear performance decrease, while Relational Masking and TGNN provide further consistent gains. These results attribute the overall improvement to the complementary effects of relation-aware structure induction and object-centric memory transition, which jointly capture evolving object relations and support future state prediction.

Table 3: Overall component ablation with VideoSAUR representations.

Variants	DyG	Rel. Mask	TGNN	Mem. Pred.	Avg. (%)
C-JEPA	×	×	×	×	89.40
w/o DyG	×	×	×	✓	90.63
w/o Rel. Mask	✓	×	✓	✓	90.94
w/o TGNN	✓	✓	×	✓	90.97
w/o Mem. Pred.	✓	✓	✓	×	90.21
Full WAG	✓	✓	✓	✓	**91.37**

Table 4: Comparison of different masking strategies with SAVi representations.

Masking Strategy	Avg.	Counter.	Predict.
Random [w/o DyG.]	83.88	60.19	77.25
Random [w/ DyG.]	92.28 (+8.40)	73.75 (+13.56)	91.14 (+13.89)
Relational Centrality	**92.72** (+8.84)	**75.45** (+15.26)	**91.99** (+14.74)
Temporal Dynamics	91.95 (+8.07)	73.51 (+13.32)	89.71 (+12.46)

In-Depth Ablation on Latent Dynamic Graph Construction. To demonstrate the effectiveness of latent dynamic graph construction in **RQ2**, we compare our constructed KNN latent dynamic graphs in WAG with two uniformly sampled graph variants. Specifically, uniform graph (IID) samples K neighbors independently, allowing duplicates and self-loops, while uniform graph (subset) samples K distinct neighbors without replacement and excludes self-loops. As shown in Figure 3, our WAG consistently outperforms both uniformly sampled graph variants across all CLEVRER question types. These results indicate that the gain does not only come from introducing graph connectivity, but also from constructing informative relations in the latent space.

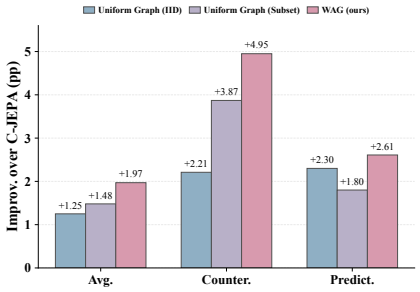


Figure 3: Performance improvement of different latent dynamic graph construction strategies over C-JEPA.

In-Depth Ablation on Relation-Aware Object Masking. Table 4 compares different object masking strategies under SAVi representations in **RQ2**. Introducing dynamic graph information, i.e., random [w/ DyG.], consistently improves random masking over random [w/o DyG.], i.e., the C-JEPA setting. Among the graph-based strategies, relational centrality achieves the best performance across all reported metrics, improving the average per-question accuracy by 8.84 percentage points under SAVi representations. Temporal dynamics also provides consistent gains over the random [w/o DyG.], while relational centrality further demonstrates the benefit of selecting objects according to their relational importance.

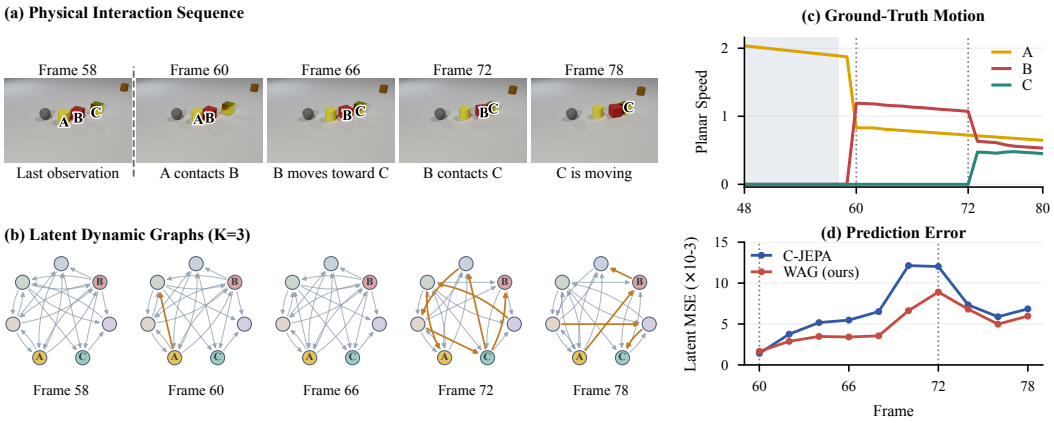


Figure 4: Visualization of physical interaction modeling on CLEVRER.(a) Physical interaction from the last observed to future frames. (b) Latent dynamic graphs constructed by WAG with $K = 3$, showing evolving object relations. A/B/C correspondences are inferred. (c) Ground-truth object motion changes. (d) Latent prediction errors of C-JEPA and WAG, with lower errors achieved by WAG during interaction-driven transitions.

3.4 In-Depth Analysis of WAG

Analysis of Physical Interaction Modeling. To answer **RQ3**, we present Figure 4, a CLEVRER sequence with a two-step collision chain, where A first strikes B and B subsequently strikes C within the prediction horizon. Although the model only observes the frames before these interactions, the latent graph changes when the physical relations evolve, especially around the second collision. Compared with C-JEPA, WAG reduces the mean latent MSE ($\times 10^{-3}$) from 6.66 to 4.83, while lowering the peak error from 12.1 to 8.9. This suggests that the gain mainly comes from dynamic relation updates, which better capture interaction-driven state changes and future dynamics.

Analysis of the Number of Masked Objects M and Graph Neighborhood Size K. Figure 5 studies the sensitivity of WAG to the number of masked objects M and graph neighborhood size K in **RQ4**. Overall, the performance remains relatively stable across different parameter combinations. Moderate neighborhood sizes (K=3) generally achieve stronger performance, while the preferred masking number varies slightly across representations and evaluation metrics. In particular, VideoSAUR achieves its highest average accuracy of 91.37% at K=3, M=2. More results are in Appendix A.4.

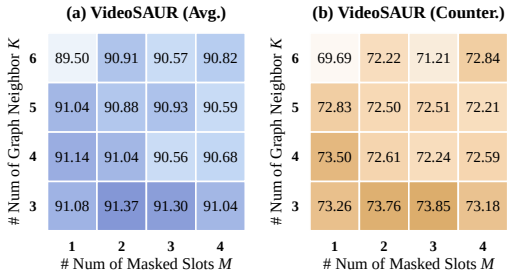


Figure 5: Analysis of the number of masked objects M and graph neighborhood size K.

Analysis of Running Costs. For **RQ4**, we report the training time and peak GPU memory. Our proposed WAG is substantially more efficient than C-JEPA and OC-JEPA, requiring only about 143 seconds per epoch compared with 341.4 seconds for C-JEPA, i.e., approximately 2.4× faster training. It also reduces peak GPU memory from 6712 MiB to 410 MiB, achieving up to a 16.4× reduction. More detailed running costs, including GFLOPs, are provided in Appendix A.4.

4 Conclusion

In this work, we introduce **W**orld-**A**s-**G**raph (**WAG**), a graph-based object-centric world model that explicitly captures object relations and their temporal evolution in latent space. WAG integrates relation-aware structure induction with object-centric memory transition, allowing relational information and historical object states to jointly support future prediction. Experiments on visual reasoning and robotic manipulation tasks demonstrate the consistent improvements of our proposed WAG. In-depth ablation studies and analysis, along with physical interaction analysis, further highlight the effectiveness of representing the latent world as dynamic graphs. More broadly, WAG provides a general framework for connecting object-centric representation learning with relational graph modeling, offering a step toward more structured and interaction-aware world models.

AI USE STATEMENT

In this work, we used generative AI tools for editing and improving the readability of the manuscript, refining the presentation of scientific figures, and supporting software code development and debugging. The research ideas, proposed methodology, experimental design, and interpretation of the experimental results were developed and conducted by the authors. All AI-assisted text was reviewed and revised by the authors. No AI-generated content was directly adopted without author verification. We take responsibility for the final content of this work, including text, claims, code, figures, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personal or sensitive data, or other ethical concerns requiring specific consideration.

REPRODUCIBILITY STATEMENT

We provide the main architectural details and learning objectives of WAG in the Method section. The datasets, evaluation protocols, baseline settings, and partial implementation details are described in the Experiment section. Additional hyperparameter settings, model configurations, and implementation details required for reproducing the reported results are provided in the Appendix.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Jiaxin Bai and Jiaxuan Xiong. Temporal-distance jepa: Plan-aware representation learning for latent world model predictive control. *arXiv preprint arXiv:2607.25337*, 2026.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2(3):5, 2018.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35101–35113, June 2026.

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

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. Do we really need complicated model architectures for temporal networks? *arXiv preprint arXiv:2302.11636*, 2023.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. 2024:2632–2652, 2024.

David Ding, Felix Hill, Adam Santoro, Malcolm Reynolds, and Matt Botvinick. Attention over learned object embeddings enables complex visual reasoning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:9112–9124, 2021.

Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, et al. Understanding world or predicting future? a comprehensive survey of world models. *ACM Computing Surveys*, 58(3):1–38, 2025.

Fan Feng, Phillip Lippe, and Sara Magliacane. Learning interactive world model for object-centric reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Tao Feng, Yexin Wu, Guanyu Lin, and Jiaxuan You. Graph world model. In *Forty-second International Conference on Machine Learning (ICML)*, 2025b.

ZhengZhao Feng, Rui Wang, TianXing Wang, Mingli Song, Sai Wu, and Shuibing He. A comprehensive survey of dynamic graph neural networks: Models, frameworks, benchmarks, experiments and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2025c.

Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Focus: object-centric world models for robotic manipulation. *Frontiers in Neurorobotics*, 19:1585386, 2025.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Yiming He, Xiang Li, Zhongying Zhao, Haobing Liu, Peilan He, and Yanwei Yu. Maskdgnn: Self-supervised dynamic graph neural networks with activeness-aware temporal masking. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2892–2900, 2025.

Ming Jin, Yuan-Fang Li, and Shirui Pan. Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:19874–19886, 2022.

Thomas Kipf, Elise Van der Pol, and Max Welling. Contrastive learning of structured world models. *International Conference on Learning Representations (ICLR)*, 2020.

Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021.

David Klindt, Yann LeCun, and Randall Balestriero. When does lejepa learn a world model? *arXiv preprint arXiv:2605.26379*, 2026.

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

Anson Lei, Bernhard Schölkopf, and Ingmar Posner. Spartan: A sparse transformer world model attending to what matters. *Advances in Neural Information Processing Systems (NeurIPS)*, 38: 154089–154114, 2026.

Jintang Li, Ruofan Wu, Xinzhou Jin, Boqun Ma, Liang Chen, and Zibin Zheng. State space models on temporal graphs: A first-principles study. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:127030–127058, 2024.

Jiawei Liu, Senqiao Yang, Mingjun Wang, Yu Wang, and Bei Yu. Graph world models: Concepts, taxonomy, and future directions. *arXiv preprint arXiv:2604.27895*, 2026.

Malte Mosbach, Jan Niklas Ewertz, Angel Villar-Corrales, and Sven Behnke. Sold: Slot object-centric latent dynamics models for relational manipulation learning from pixels. *arXiv preprint arXiv:2410.08822*, 2024.

Heejeong Nam, Quentin Le Lidec, Lucas Maes, Yann LeCun, and Randall Balestriero. Causal-jepa: Learning world models through object-level latent interventions. *arXiv e-prints*, pp. arXiv–2602, 2026.

Yosuke Nishimoto and Takashi Matsubara. Object-centric world models for causality-aware reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24585–24593, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Fachrina Dewi Puspitasari, Chaoning Zhang, Joseph Cho, Adnan Haider, Noor Ul Eman, Omer Amin, Alexis Mankowski, Muhammad Umair, Jingyao Zheng, Sheng Zheng, et al. Sora as a world model? a complete survey on text-to-video generation. *arXiv preprint arXiv:2403.05131*, 2024.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Ayumu Saito, Prachi Kudeshia, and Jiju Poovvancheri. Point-jepa: A joint embedding predictive architecture for self-supervised learning on point cloud. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7348–7357. IEEE, 2025.

Xinyuan Song and Zekun Cai. Understanding rollout error in graph world models. *arXiv preprint arXiv:2606.27780*, 2026.

Ludovic Tuncay, Etienne Labbé, Emmanouil Benetos, and Thomas Pellegrini. Audio-jepa: Joint-embedding predictive architecture for audio representation learning. *arXiv preprint arXiv:2507.02915*, 2025.

Wei Wang, Yaosen Chen, Han Yang, Yuegen Liu, Mingli Luo, Xinxin Jiao, Xuming Wen, and Ming Liu. A structural dynamics graph world model: Unified modeling, constrained rollout, and interpretable calibration. *arXiv preprint arXiv:2608.08689*, 2026a.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026b.

Zizhao Wang, Kaixin Wang, Li Zhao, Peter Stone, and Jiang Bian. Dyn-o: Building structured world models with object-centric representations. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:121788–121812, 2025.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:68082–68119, 2024.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. *arXiv preprint arXiv:2210.05861*, 2022.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

Fuxiang Yang, Donglin Di, Lulu Tang, Xuancheng Zhang, Lei Fan, Hao Li, Wei Chen, Tonghua Su, and Baorui Ma. Chain of world: World model thinking in latent motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6675–6684, 2026.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:67686–67700, 2023.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:61514–61545, 2023.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:24195–24228, 2026.

Yanping Zheng, Lu Yi, and Zhewei Wei. A survey of dynamic graph neural networks. *Frontiers of Computer Science*, 19(6):196323, 2025.

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

A APPENDIX

This is the Appendix of the submission: **World-As-Graph: Relational World Modeling Through Latent Space Graphs**. In this appendix, we include more details of: (1) Related work, (2) Theoretical analysis, (3) Algorithm, and (4) Experimental details and comprehensive results.

A.1 RELATED WORK

Joint-Embedding Predictive Architectures. As a common architecture for self-supervised representation learning, the joint-embedding predictive architecture (JEPA) (LeCun et al., 2022; Assran et al., 2023; Klindt et al., 2026) aims to train an encoder and a predictor based on observations to learn informative representations of inputs in a latent space, so that it could forecast future world states from context and actions by capturing the input relationships (Klindt et al., 2026; Bai & Xiong, 2026; Saito et al., 2025; Tuncay et al., 2025). Based on this architecture, JEPA has been applied on different modalities, such as Point-JEPA (Saito et al., 2025) on point clouds and Audio-JEPA (Tuncay et al., 2025) on audio signals. LeJEPA (Balestriero & LeCun, 2025) further studies that the latent prediction objectives can be trained in a scalable manner without relying on heuristics. Moreover, different research have been studied for dynamic scenes, the predictor is required to forecast the latent states of future observations. Temporal-Distance JEPA (Bai & Xiong, 2026) shapes the latent space with a temporal distance, where these trained representations are shaped to be meaningful for downstream planning. Chain-of-World predicts (Yang et al., 2026) the future through a chain of latent motion states. Causal ideas have also been introduced into the objective, Causal-JEPA (Nam et al., 2026) masks the latent states, forcing each masked object to be predicted from the remaining objects, although which objects are masked is still decided at random.

Graph World Models. Recent graph world models mainly explore *world modeling for graphs*, where graph-structured states are treated as the environment representation for prediction, simulation, and planning (Liu et al., 2026; Feng et al., 2025b). For example, GWM (Feng et al., 2025b) provides a unified framework for modeling graph-structured and multimodal states through message passing and action nodes. More recent studies further investigate rollout errors over graph trajectories (Song & Cai, 2026) and incorporate explicit structural mechanisms and constraints into graph-based dynamics (Wang et al., 2026a). In these methods, graph structure mainly serves as the state space on which world dynamics are modeled. A related line of work introduces graph structures into object-centric world models to represent interactions among visual entities. C-SWM (Kipf et al., 2020) represents object states and their relations through a graph neural network, while FIOC-WM (Feng et al., 2025a) starts learning interaction structures between object-centric representations. Different from these studies, our work focuses on *graphs for world modeling*: latent dynamic graphs are introduced as relational inductive biases to guide object-centric predictive representation learning. In particular, WAG uses evolving graph structures not only to represent object interactions, but also to guide relational masking and object-level memory transition, encouraging future prediction to exploit both inter-object dependencies and temporal dynamics.

Dynamic Graph Representation Learning. Dynamic graphs can be widely applied in many real-world systems, where the graph topology and node attributes keep changing with time (Feng et al., 2025c; Zheng et al., 2025). The key to dynamic graph representation learning is to capture the relations and structure across different time steps. In this context, dynamic graph neural networks (DGNNs) become powerful tools to learn temporal entity interactions and representations in the real world. Typically, TGAT (Xu et al., 2020) combines time encoding with temporal attention over neighbors. TGN (Rossi et al., 2020) chooses to maintain a recurrent memory state for each graph node and keep updating the state with each interaction. NeurTWs (Jin et al., 2022) effectively capture each temporal node while preserving temporal constraints by conducting spatiotemporal-biased random walks. More recent models simplify the architecture. GraphMixer (Cong et al., 2023) shows that a simple MLP-based encoder is already competitive, while DyGFormer (Yu et al., 2023) learns from the neighbor sequence of each node with a transformer. GraphSSM (Li et al., 2024) models the evolution of temporal graphs with a state space formulation. MaskDGNN (He et al., 2025) masks the temporal edges in a self-supervised manner. This line of research is closely related to the temporal GNN component in WAG, while our goal is not general-purpose dynamic graph representation learning.

Algorithm 1 WAG Training

Require: Slot sequence $\{\mathbf{S}_t\}_{t\in\mathcal{T}}$ from f_{Φ^*}, $\mathcal{T} = \mathcal{T}_{\text{hist}} \cup \mathcal{T}_{\text{pred}}$; history/horizon lengths T_h, T_p; graph degree K; mask budget M;
masking policy $\pi_{\text{mask}} \in \{\text{Relational Centrality}, \text{Temporal Dynamics}\}$; loss weights $\lambda_{\text{mask}}, \lambda_{\text{futu}}$;
Ensure: Learned parameters (θ, ω) of WAG (best checkpoint by validation loss)

repeat
  Sample $\{\mathbf{S}_t\}_{t\in\mathcal{T}}$; let $t_0 = t - T_h + 1$
  $G_t \leftarrow \Gamma_K^{\text{sim}}(\mathbf{S}_t)$ for each $t \in \mathcal{T}_{\text{hist}}$ Eq. (1)
  Compute c_i Eq. (2) or d_i Eq. (3) per π_{mask}; $\mathcal{I}_{\text{mask}} \leftarrow \arg\text{Top-}M$
  $G_t^\dagger \leftarrow (\mathcal{V}_t, \mathcal{E}_t, \mathbf{S}_t^\dagger)$ for $t \in \mathcal{T}_{\text{hist}} \setminus \{t_0\}$, masking $\mathbf{s}_t^i$, $i \in \mathcal{I}_{\text{mask}}$; keep G_{t_0} fully visible
  $\mathbf{H}_{t_0} \leftarrow \mathbf{S}_{t_0}$
  for $t \in \mathcal{T}_{\text{hist}} \setminus \{t_0\}$ **do**
    $\mathbf{H}_t \leftarrow \text{MemGRU}_{\theta_m}\big(\text{TGNN}_{\theta_g}(G_t^\dagger), \mathbf{H}_{t-1}\big)$ Eq. (4)
    $\widehat{\mathbf{S}}_t^{\text{mask}} \leftarrow g_{\text{mask}}^{\epsilon}(\mathbf{H}_t, \mathcal{I}_{\text{mask}})$ Eq. (5)
  end for
  for $t \in \mathcal{T}_{\text{pred}}$ **do**
    $\widehat{G}_{t-1} \leftarrow \Gamma_K^{\text{sim}}(\mathbf{H}_{t-1})$; $\mathbf{H}_t \leftarrow f_{\text{pred}}^{\omega}(\widehat{G}_{t-1}, \mathbf{H}_{t-1})$; $\widehat{\mathbf{S}}_t^{\text{futu}} \leftarrow g_{\text{proj}}^{\eta}(\mathbf{H}_t)$ Eq. (6)
  end for
  $\mathcal{L}_{\text{mask}}, \mathcal{L}_{\text{futu}} \leftarrow$ Eq. (7); $\mathcal{L}(\theta, \omega) \leftarrow \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{futu}}\mathcal{L}_{\text{futu}}$ Eq. (8)
  $(\theta, \omega) \leftarrow \text{AdamW}\big((\theta, \omega), \nabla_{\theta,\omega}\mathcal{L}\big)$
  if $\mathcal{L}_{\text{val}} < \mathcal{L}_{\text{best}}$ **then**
    $(\theta_{\text{best}}, \omega_{\text{best}}) \leftarrow (\theta, \omega)$; $\mathcal{L}_{\text{best}} \leftarrow \mathcal{L}_{\text{val}}$
  end if
until converged
return $(\theta_{\text{best}}, \omega_{\text{best}})$

A.2 Theoretical Analysis

During autoregressive rollout, WAG constructs future latent graphs from evolving object-level memory states without access to future observations. Consequently, even small memory errors can change neighbor selection, alter relational message passing, and affect subsequent predictions. This coupling raises a central question:

> *Under what conditions do memory perturbations preserve KNN neighborhoods relative to a reference trajectory, and how do these errors accumulate across rollout steps?*

To answer this question, we first characterize neighborhood stability through the cosine-similarity margin at the KNN selection boundary. We then derive a memory-error bound that connects this margin condition with transition sensitivity and single-step prediction residuals.

We analyze how memory perturbations affect Γ_K^{sim} and f_{pred}^{ω} in Eq. (6). Let $\mathbf{H}_t$ denote the rollout memories and $\mathbf{H}_t^\star$ a reference memory trajectory for analysis, with all node norms bounded below by $r > 0$. Therefore, we define

$$e_t = \|\mathbf{H}_t - \mathbf{H}_t^\star\|_{\max}, \qquad \|\mathbf{H}\|_{\max} = \max_i \|\mathbf{h}^i\|_2. \tag{9}$$

For $1 \le K < N-1$, let $w_{i,(k)}^{t,\star}$ denote the k-th largest value of $\cos(\mathbf{h}_t^{i,\star}, \mathbf{h}_t^{j,\star})$ over $j \neq i$. We define the neighborhood margin as

$$\gamma_t = \min_i \left(w_{i,(K)}^{t,\star} - w_{i,(K+1)}^{t,\star} \right). \tag{10}$$

Normalized node discrepancies are at most $2e_t/r$, so each cosine similarity changes by at most $4e_t/r$. Therefore, $e_t < r\gamma_t/8$ preserves neighbor sets between predicted and reference graphs, without requiring constant topology over time or unchanged edge weights. Assume the memory update in Eq. (6) is uniformly L-Lipschitz under $\|\cdot\|_{\max}$ for fixed neighbor indices and recomputed cosine weights, with reference residual as:

$$\left\| \mathbf{H}_{t+1}^\star - f_{\text{pred}}^{\omega}\left(\Gamma_K^{\text{sim}}(\mathbf{H}_t^\star), \mathbf{H}_t^\star\right) \right\|_{\max} \le \delta. \tag{11}$$

Let $T = \max \mathcal{T}_{\text{hist}}$. If the margin condition holds throughout the rollout, then, for $1 \le h \le T_p$, we have:

$$e_{t+1} \le L e_t + \delta, \qquad e_{T+h} \le L^h e_T + \delta \sum_{j=0}^{h-1} L^j. \tag{12}$$

The triangle inequality yields the one-step bound; induction gives the multi-step bound. Thus, neighborhood separation prevents perturbation-induced rewiring, while transition sensitivity controls error accumulation. For $L < 1$, the bound remains uniform over the horizon. These are sufficient analytical conditions, not guarantees provided by recurrent memory alone.

A.3 Algorithms

Algorithm 1 summarizes the overall training procedure of our proposed WAG.

A.4 Experimental Details and Comprehensive Results

Dataset Overview. PushT (Chi et al., 2025) and CLEVRER (Yi et al., 2019) represent two distinct downstream tasks of WAG used in this paper. CLEVRER is a discrete video question-answering(VQA) and causal reasoning task, where an ALOE-style VQA head is used to classify four types of questions: *counterfactual questions*, asking how the scene would evolve if a certain object is removed; *explanatory questions*, asking which observed event is responsible for a given event; *predictive questions*, asking which event will happen after the video ends; *descriptive questions*, asking about the objects and the events in the observed frames. Performance on CLEVRER is evaluated using question-answering accuracy.

Table A1: Dataset statistics.

Property	PushT	CLEVRER
Train split	18,685 trajectories	10,000 videos
Val / test split	21 trajectories	5,000 / 5,000 videos
Object slots	4	7
Task type	2D pushing manipulation	Video QA / causal reasoning
Downstream head	CEM planner	ALOE-style
Evaluation metric	Success rate	QA accuracy

In contrast, PushT is a continuous control and planning task, which is inferred with the Cross-Entropy Method (CEM) over the learned world model, and performance is measured by task success rate. At every planning step, CEM samples a population of candidate action sequences and scores them by the learned world model's predicted cost to the goal. An episode is deemed successful if, within this budget, the agent-block configuration matches the goal state within a position error of 20 pixels and an orientation error of $\pi/9$ radians. Performance on PushT is evaluated using the success rate. The dataset statistics are summarized in Table A1.

Baselines. For the *visual reasoning* task, specifically, we consider two baseline methods: (1) OC-JEPA (Nam et al., 2026), which predicts the future object slots from a fully observed history; and (2) Causal-JEPA (C-JEPA) (Nam et al., 2026), which additionally masks a subset of object slots within the observed history and jointly trains the model to reconstruct the masked slots while predicting the future object slots. For VQA evaluation, all compared methods use the same pretrained encoder, rollout procedure, and downstream models to ensure a controlled and fair comparison.

For the *robotic manipulation* task, we take the following baseline models: (1) DINO-WM (Zhou et al., 2024), an autoregressive predictor that predicts over patch-level representations; (2) DINO-WM-Reg. (Darcet et al., 2024), a variant that uses a DINOv2 backbone while retaining the same patch-based predictor, thereby isolating the effect of the representation; (3) OC-DINO-WM (Nam et al., 2026), which replaces DINO-WM's patch embeddings with object-centric slots; (5) OC-JEPA and C-JEPA (Nam et al., 2026), which uses the same object-slot masking objectives introduced above and evaluated here on planning. To maintain comparability, all baselines start from DINOv2 embeddings (Oquab et al., 2023) and differ only in predictor training.

Implementation Details. For visual reasoning over the CLEVRER dataset and robotic manipulation over the PushT dataset, WAG is trained on a pretrained object-centric encoder. VideoSAUR is used for both datasets, while SAVi is additionally included for comparison on CLEVRER. All models are optimized with AdamW. For downstream evaluation, we select the checkpoint with the lowest

Table A2: Implementation details and hyperparameters for the CLEVRER experiments.

Symbol	Value	Description
WAG *hyperparameters*		
T_h	6	history window length
T_p	10	prediction horizon
K	3/4	DyG Builder neighbors
M	2/4	mask budget
Δ_{frame}	2	frame skip
λ_{mask}	1	masked-history loss weight
λ_{futu}	0.25/0.5	future-prediction loss weight
α_{pred}	5×10^{-4}	predictor learning rate
n_{batch}	256	batch size
E	60	predictor training epochs
	best validation loss	checkpoint selection
Downstream VQA (ALOE)		
E_{ALOE}	400	ALOE training epochs
α_{ALOE}	1×10^{-3}	ALOE learning rate
	fp16	precision
	0	random seed

Table A3: Implementation details and hyperparameters for the PushT experiments.

Symbol	Value	Description
WAG *hyperparameters*		
T_h	3	history window length
T_p	3	prediction horizon
K	2	DyG Builder neighbors
M	1	mask budget
Δ_{frame}	5	frame skip
D_{proprio}	128	proprioception embedding dim
D_{act}	128	action embedding dim
α_{pred}	5×10^{-4}	predictor learning rate
α_{proprio}	1×10^{-4}	proprioception encoder learning rate
α_{act}	5×10^{-4}	action encoder learning rate
n_{batch}	256	batch size
E	30	predictor training epochs
Planning		
T_h	3	history size
n_{seed}	$\{0, 1, 2\}$	evaluation seeds

Table A4: Comparison of masking strategies for **WAG$_{\text{I}}$** vs. **WAG$_{\text{B}}$** on the VideoSAUR encoder (all runs at $\lambda_{\text{futu}} = 1$) and $\lambda_{\text{mask}} = 1$.

Masking	Model	Average per que. (%)	Counterfactual (%)		Explanatory (%)		Predictive (%)	
			per opt.	per que.	per opt.	per que.	per opt.	per que.
Random	WAG$_{\text{I}}$	90.94	90.20	72.83	97.69	93.54	93.97	88.75
	WAG$_{\text{B}}$	**91.14**	90.31	72.76	97.71	93.64	**94.21**	**89.34**
	Δ (B−I)	+0.20	+0.11	−0.07	+0.02	+0.10	+0.24	+0.59
Relational Centrality	WAG$_{\text{I}}$	**90.79**	90.06	**72.22**	**97.79**	**93.96**	92.92	86.81
	WAG$_{\text{B}}$	90.85	89.96	72.20	97.31	92.80	**93.52**	**87.94**
	Δ (B−I)	+0.06	−0.10	−0.02	−0.48	−1.16	+0.60	+1.13
Temporal Dynamics	WAG$_{\text{I}}$	90.47	89.14	70.16	96.95	91.74	93.41	87.71
	WAG$_{\text{B}}$	**90.94**	**90.23**	**72.51**	**97.55**	**93.31**	**93.86**	**88.59**
	Δ (B−I)	+0.47	+1.09	+2.35	+0.60	+1.57	+0.45	+0.88

validation loss rather than the checkpoint from the final training iteration. In the following, we provide detailed parameter settings for each experimental component. Unless otherwise specified, all experiments follow the default configuration in Table A2 and Table A3.

For PushT, the four object slots are augmented with two auxiliary tokens encoding the proprioceptive state and action, respectively. The state and action are separately projected into the same D-dimensional latent space as the object slots and appended to the object representations, resulting in six latent tokens in total. During CEM planning, each candidate action is encoded as the action token to condition the corresponding future transition, while the proprioception token provides the current robot state.

In Table 1 of the main submission, we report the best-performing results for WAG$_{\text{I}}$ and WAG$_{\text{B}}$ when using Relational Centrality as the masking mechanism. For VideoSAUR, we set $\lambda_{\text{futu}} = 0.25$; the reported WAG$_{\text{I}}$ results use K=4 and M=4, while WAG$_{\text{B}}$ uses K=3 and M=2. For SAVi, we set $\lambda_{\text{futu}} = 0.5$; WAG$_{\text{B}}$ uses K=4 and M=2, whereas the reported WAG$_{\text{I}}$ results use K=4 and M=4. ALOE is trained for 400 epochs. When analyzing the effect of λ_{futu} on performance, we trained ALOE for only 100 epochs as a reference due to computational constraints. All experiments are run on a single NVIDIA H20 GPU.

Instance-Wise vs. Batch-Shared Masking. Under identical parameter settings, we also evaluate the effects of instance-wise masking and batch-shared masking across different masking strategies, with the results shown in Table A4. For the random mask strategy, WAG$_{\text{B}}$ also achieves small but consistent improvements over the instance-wise masking variant WAG$_{\text{I}}$, although the gains are substantially small.

Table A5: Overall ablation study of key components in our proposed WAG with VideoSAUR encoder. Counter. and Predict. denote per-question accuracy for counterfactual and predictive questions, respectively.

	Relation-Aware Structure Induction		Object-Centric Memory Transition		Performance (%) ↑				
Variants	DyGraph	Rel. Mask	TGNN	Mem. Pred.	Avg. per que.	Counter.	Explan.	Predict.	Descrip.
C-JEPA	×	×	×	×	89.40	68.81	90.74	86.93	92.84
w/o DyGraph	×	×	×	✓	90.63	72.68	93.38	89.15	93.36
w/o Rel. Mask	✓	×	✓	✓	90.94	72.83	93.54	88.75	93.75
w/o TGNN	✓	✓	×	✓	90.97	73.42	93.52	90.33	93.59
w/o Mem. Pred.	✓	✓	✓	×	90.21	71.50	92.46	85.55	93.34
Full WAG (ours)	✓	✓	✓	✓	**91.37**	**73.76**	**94.13**	**89.54**	**94.05**

Table A6: Comparison of different object masking strategies under VideoSAUR and SAVi representations using per-question accuracy. Values in parentheses denote absolute improvements over Random [w/o DyG.] (i.e., C-JEPA) in percentage points, where w/o DyG. and w/ DyG. indicate without and with latent dynamic graph construction, respectively.

Masking Strategy	*VideoSAUR Encoder*			*SAVi Encoder*		
	Avg. per. que. (%)	Counter. (%)	Predict. (%)	Avg. per. que. (%)	Counter. (%)	Predict. (%)
Random [w/o DyG.]	89.40	68.81	86.93	83.88	60.19	77.25
Random [w/ DyG.]	90.94 (+1.54)	72.83 (+4.02)	88.75 (+1.82)	92.28 (+8.40)	73.75 (+13.56)	91.14 (+13.89)
Relational Centrality	91.37 (+1.97)	73.76 (+4.95)	89.54 (+2.61)	92.72 (+8.84)	75.45 (+15.26)	91.99 (+14.74)
Temporal Dynamics	90.47 (+1.07)	70.16 (+1.35)	87.71 (+0.78)	91.95 (+8.07)	73.51 (+13.32)	89.71 (+12.46)

In contrast, WAG_B shows the clearest and most consistent advantage under the temporal dynamics strategy. In particular, it improves counterfactual per-question accuracy by 2.35 points and explanatory per-question accuracy by 1.57 points, and only a marginal decrease of 0.05 on the descriptive category. This suggests that batch-shared masking, as used in WAG_B, is particularly effective when temporal changes drive the masking strategy.

Meanwhile, the results under relational centrality indicate that the relative benefit of batch-shared versus instance-wise masking depends on the masking strategy. Although WAG_B achieves a higher overall score by 0.06 points, WAG_I performs substantially better on explanatory per-question accuracy, with a gain of 1.16 points. This suggests that, unlike temporal-change masking, degree-based masking does not consistently benefit from batch-shared masking.

Ablation on Graph Connectivity and MLP Capacity. We further analyze the effects of graph connectivity and model capacity on PushT by performing an ablation over $K \in \{2, 4\}$ and the MLP model size $\in \{2048, 4096\}$, with the results shown in Figure A1. Each experiment is evaluated with three seeds and 50 episodes per seed. The results show that graph K is the dominant factor. Increasing K from 2 to 4 consistently reduces the mean success rate by approximately 15 percentage points, from 90.7 to 76.7 with the smaller MLP configuration and from 88.7 to 73.3 with the larger MLP configuration. In contrast, increasing the MLP capacity by enlarging its hidden dimension from 2048 to 4096 results in only a modest but consistent performance drop under both K settings, indicating that model capacity is already sufficient at width 2048 and further expansion provides no benefit.

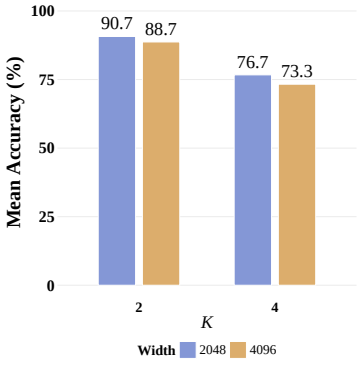


Figure A1: Analysis of graph neighborhood size K and MLP model size with width.

Computational Complexity. WAG has a history window of length T_h and prediction horizon T_p, with N object slots of dimensionality D and $K < N$ nearest neighbors per node. Thus, the cost of our predictor can be dominated by (a) KNN graph construction via cosine similarity, $O(N^2 D)$;

Table A7: VQA accuracy comparison of different random-graph sampling strategies and WAG$_{\text{B}}$ using the VideoSAUR encoder.

Model	Average	Counterfactual (%)		Explanatory (%)		Predictive (%)		Descriptive (%)
	per que. (%)	per opt.	per que.	per opt.	per que.	per opt.	per que.	
C-JEPA	89.40	88.67	68.81	96.62	90.74	93.03	86.93	92.84
Random Graph (IID)	90.65	89.52	71.02	97.28	92.60	94.24	89.23	93.77
Random Graph (Subset)	90.88	90.24	72.68	97.55	93.27	93.94	88.73	93.74
WAG$_{\text{B}}$ (ours)	**91.37**	**90.55**	**73.76**	**97.84**	**94.13**	**94.43**	**89.54**	**94.05**

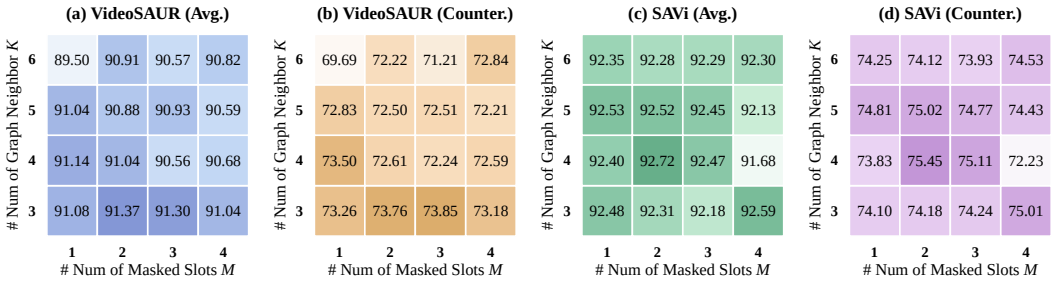
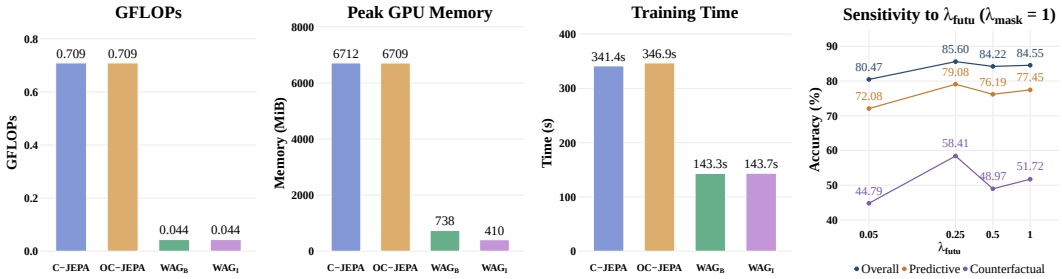


Figure A2: Analysis of the number of masked objects M and graph neighborhood size K. Average per-question and counterfactual VQA accuracy (%) are reported under VideoSAUR and SAVi representations.



(a) GFLOPs. (b) Memory usage. (c) Running time. (d) Sensitivity to λ_{futu}.

Figure A3: Computational efficiency and hyperparameter sensitivity analysis. (a) Computational cost measured in GFLOPs. (b) Memory consumption. (c) Running time. (d) Performance sensitivity to the hyperparameter λ_{futu}.

(b) graph message passing over K neighbors, $O(NKD + ND^2)$; and (c) the GRU-based memory update, $O(ND^2)$. Summed over T steps, the total complexity is

$$O\big(T(N^2D + NKD + ND^2)\big) = O\big(TND(N + K + D)\big).$$

Since $D \gg N > K$ in our setting (D=128, N=7, $K \leq 6$), this simplifies to $O(TND^2)$, linear complexity in both the number of objects N and the sequence length T. This contrasts with a joint object-temporal Transformer predictor such as C-JEPA, which attends jointly over all TN object-timestep tokens and thus incurs $O((TN)^2 D_{\text{model}} + TN\, D_{\text{model}} D_{\text{mlp}})$ quadratic in TN. This gap is empirically reflected in a 16× reduction in FLOPs per sample and 9-16× lower peak memory for our predictor relative to C-JEPA and OC-JEPA (Figure A3).

Comprehensive Results. We provide the comprehensive results in the main submissions here, including: (1) Full overall ablation study results in Table A5; (2) Full comparison of different object masking strategies in Table A6; (3) Full VQA accuracy comparison of different random-graph sampling strategies in Table A7; (4) Full analysis of the number of masked objects M and graph neighborhood size K in Figure A2; (5) Full running costs and hyperparameter analysis in Figure A3.