SyncMos: Scalable Motion Synchronisation for Multi-Agent Scene Interaction

Lingxiao Li[1,*] Dongwon Kim[1,2,*] Lingyan Ruan[1] Taesoo Kwon[2] Bin Chen[1] Taehyun Rhee[1,†]
[1]University of Melbourne [2]Hanyang University
lingxiao1@student.unimelb.edu.au; tuna831@hanyang.ac.kr; lingyan.ruan@unimelb.edu.au;
taesoo@hanyang.ac.kr; bin.chen@unimelb.edu.au; taehyun.rhee@unimelb.edu.au

Abstract

Text-guided motion generation in 3D scenes has advanced the synthesis of human–scene interactions, contributing to embodied AI, scene understanding, and virtual agent simulation. While recent studies have begun exploring multi-agent scenarios, achieving temporally synchronised interactions among multiple agents remains an open challenge. Existing methods are often limited in flexibility and scalability when handling diverse interaction contexts. We present a method that enables synchronised multi-agent interaction using a single-agent motion synthesis model through two key components: a text-guided dependency-aware story planner and a temporal synchronisation module. The story planner interprets natural language instructions into structured event sequences with temporal dependencies. Our synchronisation module, built upon time-warping control and diffusion posterior sampling, aligns interaction timing across agents without retraining. Experimental results demonstrate that the proposed framework effectively models temporal dependencies and causal order between events. Evaluations across diverse interaction types show improved temporal alignment and coherent multi-agent motion generation consistent with textual instructions.

1. Introduction

Generating coordinated human motion within complex 3D scenes is a fundamental yet challenging goal in computer vision, machine learning, and embodied AI. Recent advances in text-guided motion synthesis have enabled controllable character behaviours that respond to scene context and natural-language instructions [4, 9, 22]. While these approaches achieve promising results for single-agent motion generation, extending them to multi-agent interactions presents additional challenges.

Interactions involving multiple 3D humans or agents require not only spatial reasoning but also precise temporal coordination, where actions such as handing, receiving, or reacting depend on the causal order and synchronisation between participants. Recent work has explored the synthesis of multi-agent and human–scene interactions [5, 12, 13], showing promising progress in modelling complex collaborative behaviours. However, most existing methods are restricted to a fixed number of agents or rely on pairwise relationships, which limits scalability as new interaction or relationship configurations require retraining or model redesign. Furthermore, temporal synchronisation across agents is often under-explored, leading to timing misalignments or inconsistent interactions, such as when agents hand over or receive objects, even if individual motions appear realistic.

We introduce **SyncMos**, a scalable framework for temporally synchronised multi-agent motion generation in 3D scene interaction. SyncMos builds on a single-agent diffusion model and comprises two key components. *First*, a **text-guided dependency-aware story planner** interprets natural-language instructions into structured event sequences with temporal dependencies. *Second*, a novel **temporal synchronisation module** aligns interaction timing across agents through time-warping control and diffusion posterior sampling. By leveraging a single-agent model without retraining for specific pairwise or group relationships, SyncMos achieves scalability across varying numbers of agents, while ensuring consistent timing and coordination in complex interaction contexts.

Our main contributions are as follows:

- We introduce a **two-stage framework** that extends single-agent motion generators to scalable and temporally synchronised multi-agent human–scene interactions.
- We develop a **text-guided event planner** based on a large language model that structures natural-language instructions into temporally ordered dependencies.
- We propose a **temporal synchronisation mechanism** that achieves consistent cross-agent coordination through time-warping control and diffusion-based constraints,

*These authors contributed equally.
†Corresponding author.

without additional retraining.
The framework is model-agnostic and can be integrated with existing single-agent diffusion-based motion generators to provide scalable solutions for temporally scheduled multi-human 3D scene interaction.

2. Related Work

2.1. Human motion synthesis

Recent advances in human motion synthesis stem from improvements in dataset fidelity [14, 23] and the emergence of diffusion-based generative models [17, 21]. These models synthesise realistic motion from various inputs, including action labels [7], text [17], and 3D scene context [1, 16]. Beyond single-agent motion, recent work extends to human–human [12] and human–scene interactions [9]. **InterGen** [12] produces realistic two-person interactions but relies on fixed-size joint modelling, limiting scalability to larger groups or dynamic agent counts. **LINGO** [9] enables autoregressive motion synthesis in structured 3D scenes, generating long-horizon human–scene interactions. However, existing single-agent models, including LINGO, lack mechanisms for temporally aligned multi-agent coordination. In contrast, our work introduces **SyncMos**, a general framework that combines an LLM-based dependency-aware event planner with a diffusion-based synchronisation module, achieving scalable and temporally coherent multi-agent interaction without retraining.

2.2. LLM-based Scene-level Agent Planning

Large Language Models (LLMs) have demonstrated strong capabilities in reasoning, planning, and few-shot learning [15, 18]. Recent works have leveraged LLMs for scene-level human action planning. **Digital Life Project (DLP)** [3] constructs autonomous virtual humans with lifelike social behaviours, but its focus is on simulating digital life rather than generating coordinated motions. **Sitcom-Crafter** [5] introduces a plot-driven framework that integrates diverse interactions within 3D environments, though it remains limited to two-character scenarios. **Event-Driven Storytelling** [13] decomposes complex narratives into sequential events for scalable multi-character behaviour generation in 3D scenes, yet lacks explicit temporal dependency modelling and precise synchronisation across agents. In contrast, our framework models both sequential and parallel event dependencies and achieves cross-agent synchronisation through time-warping control.

2.3. Temporal Motion Synchronisation

Precise temporal coordination is essential for generating natural and coherent motion. Classical methods such as motion warping [20] and Laplacian optimisation [11] adjust timing in captured sequences through interpolation and constraint-based editing. Recent deep learning models introduce spatial conditioning for motion synthesis [10], yet temporal control remains more challenging since shifting one action requires globally consistent adjustment across frames. InterGen [12] learns two-character interactions without explicit timing control but still struggles to produce synchronised or temporally diverse behaviours. Our framework addresses this limitation through diffusion-based temporal control using Diffusion Posterior Sampling (DPS) [6] on a pre-trained single-character motion model.

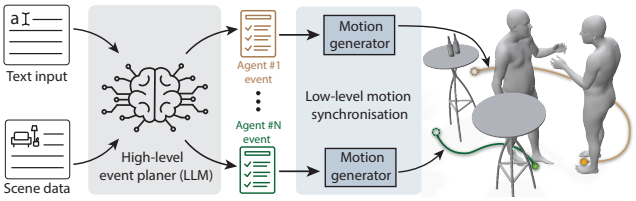

Figure 1. **Overall architecture.** The framework first interprets the user's input via a high-level planner, and low-level controller synchronizes the timing of each-character.

3. Overview

Figure 1 illustrates the overall architecture of **SyncMos**, a scalable framework for temporally synchronised multi-agent motion generation from free-form textual instructions. The framework consists of two main components: a **high-level event planner** and a **low-level motion synchronisation module**.

The high-level event planner analyses the user's textual instruction using a large language model to extract structured event descriptions with temporal and causal dependencies. It also integrates 3D scene context, such as object segmentation, spatial relations, and scene geometry to generate an ordered sequence of actions across multiple agents. The resulting plan specifies which actions each agent performs and the corresponding temporal schedule.

The low-level motion synchronisation module synthesises detailed motion sequences for each agent using a single-agent diffusion-based generative model conditioned on the planned events. To ensure temporal consistency across agents, this module incorporates a time-warping mechanism and Diffusion Posterior Sampling (DPS) [6] to align motion timing and interaction phases. Through this two-level architecture, SyncMos extends single-agent motion generation frameworks to coordinated multi-agent scenarios without retraining, providing a scalable and modular approach for generating temporally synchronised human–scene interactions.

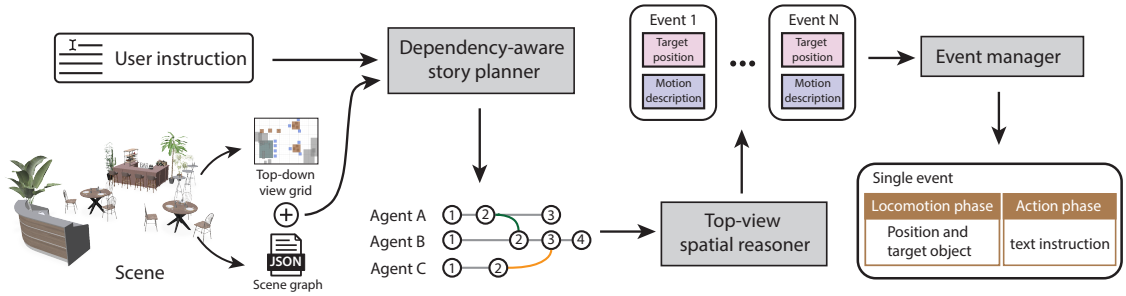

Figure 2. **The high-level event planner overview.** The high-level planner organizes dependencies while determining event locations.

4. Text-guided Event Planner

The high level text-guided event planner generates a structured event graph G from user-provided instructions T, providing explicit temporal and spatial constraints for low-level motion generation. As shown in the Figure 2, the planner consists of three sub-modules: Scene Understanding Module, Dependency-Aware Story Planner, and Top-View Spatial Reasoner.

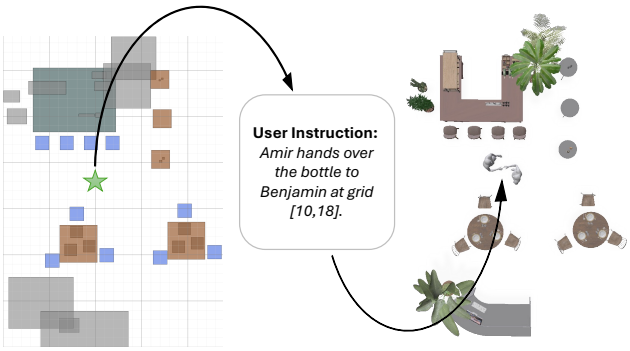

Figure 3. The top-view grid provides a unified 2D spatial coordinate system for grounding instructions. The instruction "Amir hands over the bottle to Benjamin at grid $(10, 18)$" specifies an action location on the grid (left), which corresponds to the same position in the 3D scene (right). This grounding enables spatial consistency in later motion generation stages.

4.1. Scene Understanding Module

The Scene Understanding Module extracts both semantic and geometric information from the input 3D scene S. It adopts an LLM-based scene describer [13] to capture relational semantics among objects (e.g., "the chair is near the table", "the bottle is on the counter") and to produce a textual scene description D of the the given scene.

We further introduce a top-view grid representation M that encodes spatial occupancy and navigable regions in a 2D image, providing a consistent coordinate system for subsequent spatial reasoning. The top-view grid acts as a shared spatial interface between user instructions and the motion generation process (see Fig. 3). Users can specify where an action should occur by referencing grid coordinates (e.g., "at grid $(10, 18)$"), while the Top-View Spatial Reasoner leverages the same coordinate to infer plausible spatial configurations. For example, placing two interacting characters at reachable and physically coherent distances. This design provides both intuitive user control and geometrically grounded reasoning for motion synthesis.

4.2. Dependency-Aware Story Planner

Given the scene description D from the Scene Understanding Module and the user-provided narrative text T, the planner converts high-level text into a *structured event dependency graph* that organises all actions and their temporal relations.

$$G = (E, R), \quad E = \{e_i\}_{i=1}^{N}, \quad R = R_{\text{seq}} \cup R_{\text{par}},$$

where E denotes the set of *single-actor events*, and R encodes their temporal dependencies. Each event $e_i = (\texttt{actor}_i, \texttt{event\_description}_i)$ specifies who performs the action and what it entails, for example, *"pick up bottle_1 using right hand"*. This explicit representation provides concrete cues for downstream motion generation.

To infer the dependency set R, we adopt a *few-shot, prompt-based strategy* and Chain-of-thought prompting [19]. The LLM is prompted in two stages within a single response: (1) list all required single-actor events conditioned on (D, T); and (2) infer their temporal dependencies by following structured exemplars. Two dependency types are demonstrated in the exemplars:

- **Sequential:** {`"after":e_i,"before":e_j`}, representing causal or prerequisite actions (e.g., *pick up* $\rightarrow$ *hand over*);
- **Parallel:** {`"parallel":[e_i,e_j]`}, representing synchronized interactions (e.g., giver and receiver acting simultaneously).

By exposing these templates in-context, the LLM learns to generate, in a single output, both the event list E and a machine-readable dependency set R that unifies semantic context and temporal reasoning.

We evaluate the planner's effectiveness in Sec 6.1 by comparing its dependency accuracy and event ordering with an Event-Driven Storytelling baseline [13]. Additional details are provided in the supplementary material.

4.3. Top-View Spatial Reasoner

While the Story Planner defines the semantic and temporal order of events, it does not specify where these actions occur. The Top-View Spatial Reasoner bridges this gap by grounding each abstract event into a concrete position within the scene. It grounds these textual events into concrete spatial locations within the 3D environment, providing the necessary spatial input for the subsequent motion generation module.

Given the top-down view grid and object positions, the reasoner predicts plausible target locations for each character and object interaction. It leverages spatial occupancy and accessibility constraints to ensure non-overlapping positions, continuity across consecutive events, and consistency with the physical environment (e.g., the waiter remains near the same table when serving and collecting dishes).

For each event e_i, the reasoner predicts a grounded representation

$$g_i = (\texttt{grid}_i, \texttt{action}_i, \texttt{hand\_target}_i)$$

where $\texttt{grid}_i$ indicates the 2D position on the scene map, $\texttt{action}_i$ specifies the executed action label, and $\texttt{hand\_target}_i$ optionally refers to the interacted object. This spatial grounding provides explicit positional and interaction cues for the motion generation model. Implementation details are provided in the supplementary material.

5. Temporal synchronisation model

We extend single-agent diffusion-based motion generators to produce temporally aligned multi-agent interactions guided by event-level plans. Our synchronisation framework is model-agnostic and operates atop existing auto-regressive motion models. In practice, we adopt LINGO [9] as the motion backbone, while the proposed framework can be applied to other diffusion-based generators without retraining.

The method consists of two stages: (1) *auto-regressive preliminary estimation*, which produces coarse motion sequences guided by planned events; and (2) *temporally guided refinement*, which aligns timing across agents via time-warping control and constraint-guided diffusion refinement, achieving scalable synchronised motion generation.

5.1. Auto-Regressive Preliminary Estimation

The first stage produces a coarse preliminary motion sequence $\mathcal{D}$ that follows the high-level event plan and provides a foundation for later temporal correction. The initial two poses $\mathbf{p}'^{(0)}$ and $\mathbf{p}'^{(1)}$ are fixed, and subsequent poses are generated autoregressively in fixed-length seg-

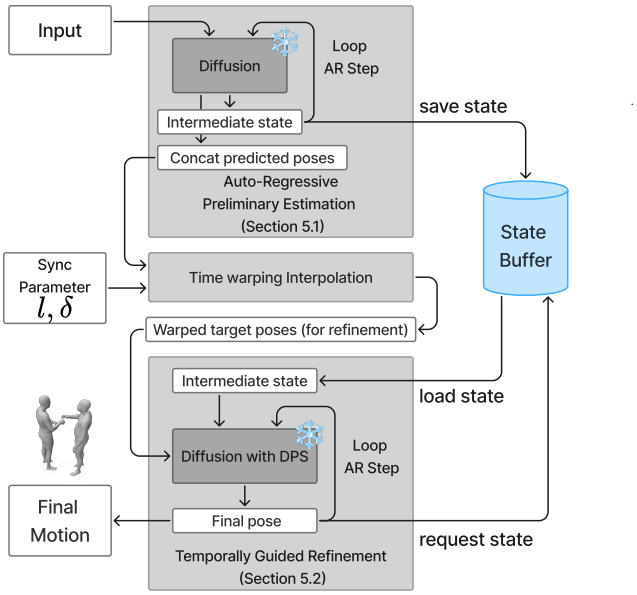

Figure 4. The temporal synchronisation model produces preliminary motion estimates for each character, which are subsequently synchronised using temporally guided refinement.

Algorithm 1 Auto-Regressive Preliminary Estimation

Require: LINGO conditioning c, total denoise steps T, intermediate step $0 < t < T$, initial prior poses $\mathbf{p}'^{(0)}, \mathbf{p}'^{(1)}$
Ensure: Intermediate state buffer $\mathcal{B}$, preliminary estimation $\mathcal{D} = \{\mathbf{p}'^{(0)}, \mathbf{p}'^{(1)}, \hat{\mathbf{p}}_0^{(2)}, \hat{\mathbf{p}}_0^{(3)}, \cdots, \hat{\mathbf{p}}_0^{(k)}\}$
$\mathcal{B} \leftarrow$ init_buffer()
$\mathcal{D} \leftarrow \{\mathbf{p}'^{(0)}, \mathbf{p}'^{(1)}\}$
while True **do**
  $\mathbf{x_T} \leftarrow \{\mathbf{p}_\mathbf{T}^{(0)}, \mathbf{p}_\mathbf{T}^{(1)}, \cdots, \mathbf{p}_\mathbf{T}^{(n)}\}$ ▷ pure noise
  $\mathbf{x_T} \leftarrow \{\mathbf{p}'^{(0)}, \mathbf{p}'^{(1)}, \cdots, \mathbf{p}_\mathbf{T}^{(n)}\}$ ▷ inpainting
  for i=T down to t+1 **do** ▷ reverse diffusion process
    $\mathbf{z} \sim \mathcal{N}(0, I)$
    $\mathbf{x_{i-1}} = \mu_\theta(x_i, i, c) + \sigma_i \cdot \mathbf{z}$
  end for
  $\hat{\epsilon} \leftarrow \mathcal{M}_\theta(\mathbf{x_t}, t, c)$
  $\hat{\mathbf{x}}_\mathbf{0} \leftarrow \text{predict}(\mathbf{x}_t, t, \hat{\epsilon})$
  $\mathcal{D} \leftarrow \text{concat}(\mathcal{D}, \hat{\mathbf{x}}_\mathbf{0})$
  $\mathcal{B} \leftarrow \text{concat}(\mathcal{B}, \{\mathbf{x_t}, T, t, c\})$ ▷ backup state
  if scheduler$(c, \hat{\mathbf{x}}_\mathbf{0})$ returns `terminate` **then**
    break
  end if
  $\mathbf{p}'^{(0)}, \mathbf{p}'^{(1)} \leftarrow \hat{\mathbf{p}}_0^{(n-1)}, \hat{\mathbf{p}}_0^{(n)}$ ▷ to next segment
end while
  return $\mathcal{D}, \mathcal{B}$

Algorithm 2 Temporally Guided Refinement

Require: preliminary estimation $\mathcal{D}$, intermediate state buffer $\mathcal{B}$, target frame l, target frame difference δ, time warp function $\tau()$

Ensure: Final motion $\mathbf{X} = \{\mathbf{p'^{(0)}}, \mathbf{p'^{(1)}}, \mathbf{p_0^{(2)}} \cdots, \mathbf{p_0^{(k)}}\}$

```
y = τ(D, l, δ)                          ▷ spline-based timewarping
X ← {}
for state in B do
    x_t, t, T, c ← state
    x_{t+t'} = add noise(x_t, t')
    for i from t + t' down to 1 do      ▷ DPS
        z ~ N(0, I)
        x̂_0 = (1/√ᾱ_i)(x_i − √(1 − ᾱ_i) ε_θ(x_i, i))
        x_{i−1} ← μ_θ(x_i, t) − λ∇_{x_i} ||y − x̂_0||^2 + σ_i · z
    end for
    X ← concat(X, x_0)
end for
    return X
```

ments (each of length n):

$$\mathcal{D} = \{\mathbf{p'^{(0)}}, \mathbf{p'^{(1)}}, \hat{\mathbf{p}}_\mathbf{0}^{(\mathbf{2})}, \hat{\mathbf{p}}_\mathbf{0}^{(\mathbf{3})}, \cdots, \hat{\mathbf{p}}_\mathbf{0}^{(\mathbf{k})}\}.$$

As shown in Algorithm 1, LINGO receives conditioning terms c that include (1) target positions relative to the current segment, (2) voxelised scene information, and (3) semantic text embeddings with timing encodings [9]. We initialise sampling from pure noise:

$$\mathbf{x_T} = \{\mathbf{p'^{(0)}}, \mathbf{p'^{(1)}}, \cdots, \mathbf{p_T^{(n)}}\},$$

and run reverse diffusion until an intermediate timestep t to obtain $\mathbf{x_t}$. Because this stage aims only to generate a usable coarse trajectory, we denote only a partial number of diffusion steps (t = 30). From the partially denoised sample, the predicted clean estimate $\hat{\mathbf{x}}_\mathbf{0}$ is obtained with the DDPM [8] reconstruction:

$$\hat{\mathbf{x}}_\mathbf{0} = \hat{x}_0(\mathbf{x_t}, t) = \frac{1}{\sqrt{\bar{\alpha}}}\left(\mathbf{x_t} - \sqrt{1-\bar{\alpha}}\,\epsilon_\theta(\mathbf{x_t}, t)\right). \quad (1)$$

To balance quality and runtime, we regulate the denoising depth using difference $\hat{\mathbf{x}}_\mathbf{0} - \mathbf{x_0}$:

$$\hat{\mathbf{x}}_\mathbf{0} - \mathbf{x_0} = \frac{\sqrt{1-\bar{\alpha}}}{\sqrt{\bar{\alpha}}}\left(\epsilon - \epsilon_\theta\left(\mathbf{x_t}, t\right)\right). \quad (2)$$

For LINGO (linear beta, $T = 100$), using $t = 30$ ($\frac{\sqrt{1-\bar{\alpha}}}{\sqrt{\bar{\alpha}}} \approx 0.3$) yields stable preliminary trajectories suitable for refinement. Unlike standard autoregressive diffusion pipelines that fully denoise each segment before moving on, we intentionally stop early and store intermediate states, including $\mathbf{x_t}$, timestep metadata, and conditioning terms, in a buffer $\mathcal{B}$. These states are later reused for temporally guided refinement.

5.2. Temporally Guided Refinement

The second pass refines the preliminary sequence $\mathcal{D}$ by enforcing global temporal consistency across agents. Simple time-warping can shift key event timings but often introduces unnatural distortions, as it does not respect the generative prior of the diffusion model. To correct timing while preserving motion realism, we treat the time-warped trajectory as a noisy temporal observation and refine it through a constraint-guided denoising process built on the stored intermediate states in buffer $\mathcal{B}$.

Timewarping. We first construct a target sequence $\mathbf{y}$ by applying spline-based timewarping to the preliminary estimate $\mathcal{D}$ (Algorithm 1, Fig. 4). The total sequence length is kept fixed, and temporal adjustments are made by pulling or pushing specific keyframes. A frame index l indicates the key event to modify, and δ defines the desired temporal offset. Pulling a frame earlier compresses preceding motion segments, while pushing it later stretches them. We use classical Motion Warping [20] to obtain an explicit but potentially noisy target trajectory $\mathbf{y}$.

Constraint-guided diffusion refinement. To enforce timing consistency while maintaining natural motion, we refine each motion segment using a gradient-guided denoising process. Given the target sequence $\mathbf{y}$, we impose a simple L2 timing constraint:

$$C(\hat{\mathbf{x}}_\mathbf{0}) = \|\mathbf{y} - \hat{\mathbf{x}}_\mathbf{0}\|^2. \quad (3)$$

For each segment, we retrieve its intermediate diffusion state from the buffer $\mathcal{B}$, which stores the partially denoised sample $\mathbf{x}_t$ and associated metadata. Before refinement, we reintroduce controlled noise via a forward diffusion step (q-sampling), allowing the model to explore feasible motion variations while remaining consistent with the diffusion prior. We then apply gradient-guided denoising to steer the reconstruction toward satisfying the timing constraint:

$$\mathbf{x_{i-1}} \leftarrow \mu_\theta(\mathbf{x_i}, i) - \lambda \nabla_{x_i} C(\hat{\mathbf{x}}_\mathbf{0}) + \sigma_i \mathbf{z}, \quad (4)$$

where μ_θ is the predicted denoised mean, σ_i is the noise variance, $\mathbf{z} \sim \mathcal{N}(0, I)$ is Gaussian noise, and λ controls the influence of the temporal constraint. This refinement step adjusts the temporal position of key events while preserving the smoothness and coherency generated by the diffusion model.

Through this two-stage process, preliminary autoregressive prediction followed by constraint-guided refinement, our system achieves global temporal synchronisation across agents without retraining the underlying motion model.

6. Results

The experimental design for **SyncMos** is motivated by the challenge of scaling multi-agent object-mediated interaction. Unlike existing methods often limited to dyadic person-to-person interaction, our goal is to demonstrate that a modular decomposition, utilizing LLM-based planners and time synchronisable motion synthesis, can achieve global coordination and dependency consistency without the need for task-specific retraining.

Consequently, We evaluate **SyncMos** at three levels: (1) the *Dependency-Aware Story Planner* against state-of-the-art and baseline LLM planners; (2) the *Temporal Synchronisation Model* via ablations and controlled timing edits (no direct prior work); and (3) the full system in an end-to-end setting to verify that planned dependencies and timings are realised in the generated motions.

6.1. Dependency-Aware Story Planner

Experiment Setup We evaluate the planner on a custom benchmark of 30 multi-character narratives spanning three scene types, *House*, *Office*, and *Restaurant* with two to five agents per scenario. Two example scenes are visualized in Fig. 6. The benchmark comprises two subsets: *Synchronisation* (15 scenarios) testing parallel multi-agent actions, and *Dependency* (15 scenarios) testing long-horizon causal chains. Each scenario includes a ground-truth event graph $G = (E, R)$ with annotated sequential and parallel relations. We compare against the Event-Driven Storytelling baseline [13], which autoregressively generates events without explicit global dependency modelling. Planner backbones include GPT-4o and Qwen-3 (235B/8B). We report Event Coverage (EC), Dependency Accuracy (DA), Passed Scenarios (PS), and Scenario Pass Rate (SPR). Further details are in Supplementary Material.

Results and Analysis Table 1 shows consistent gains over the baseline across backbones. On the *Synchronisation* subset, our planner improves dependency accuracy by up to **21.5 p.p.** (68.4 $\rightarrow$ 89.9 %, Qwen-3-235B), reflecting better handling of concurrent multi-agent actions. On the *Dependency* subset, which stresses long causal chains, dependency accuracy increases from **11.8–20.5%** to **80–97%**, and scenario pass rate rises from **near zero** to as high as **80%** (GPT-4o). Qualitatively, the planner preserves independence between unrelated sub-events, triggers dependent actions only after prerequisites (e.g., *receive* before *drink*), and yields clearer per-event role assignment.

Token usage, which reflects reasoning efficiency and runtime cost, is shown in Figure 5. In synchronisation scenarios, our planner maintains roughly 10k tokens per case with stable EC, DA, and SPR as the number of events increases. In dependency scenarios, the baseline's token usage grows rapidly beyond eight events due to redundant regeneration, whereas ours remains nearly constant, demonstrating superior scalability with narrative complexity.

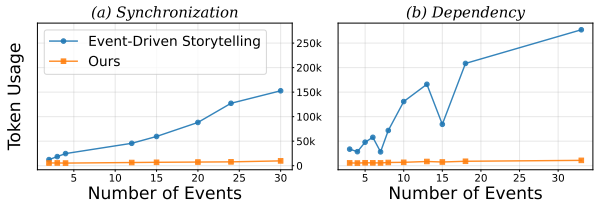

Figure 5. Token usage vs. number of events for (a) Synchronisation and (b) Dependency scenarios, comparing Event-Driven Storytelling with our planner. Each point represents the total token usage per scenario.

6.2. Temporal synchronisation model evaluation

We quantitatively evaluate the proposed temporal synchronisation model using a controlled timing adjustment experiment focused on object-grasping motions. This experiment corresponds to the most challenging dependency scenarios described in Sec. 6.1, where precise temporal alignment is essential for coordinated multi-agent actions.

Experiment Setup Scene data from the LINGO dataset are used to construct 15 distinct test cases, each beginning from a shared initial pose and targeting objects at different spatial locations. The reference *grasp timing* is defined as the frame when the wrist joint first reaches within a distance threshold of the target object.

We test timing control with offsets of ± 0.5 s, ± 1.0 s, and ± 1.5 s relative to this reference, executing each of the 15 test cases 10 times per offset condition. A trial is considered successful if the resulting temporal error is within 0.1 s of the desired offset. Temporal alignment between the preliminary estimation and the refined motion is measured using Dynamic Time Warping (DTW)[2], and a trial is marked as failed if the generated motion diverges or becomes unstable, indicated by an abnormally large DTW distance between the two sequences.

Results and Analysis Table 2 summarises the success rates for timing control under different offset magnitudes. Our model achieves over **70%** success for adjustments within ± 1.0 s and up to **88%** for smaller offsets (± 0.5 s). However, performance degrades near motion boundaries, where large offsets cause instability in the diffusion process, reducing success rates to approximately **35–40%**. These results demonstrate that the proposed synchronisation module can reliably adjust action timing without retraining, maintaining stability across moderate timing variations.

Table 1. Planner performance on Synchronisation and Dependency subsets across LLM backbones. Metrics: Event Coverage (EC), Dependency Accuracy (DA), Passed Scenarios (PS), and Scenario Pass Rate (SPR).

Model	**GPT-4o**				**GPT-4o-mini**				**Qwen-3-235B**				**Qwen-3-8B**			
Metrics	EC(%)	DA(%)	PS	SPR(%)	EC(%)	DA(%)	PS	SPR(%)	EC(%)	DA(%)	PS	SPR(%)	EC(%)	DA(%)	PS	SPR(%)
Synchronisation Test Set																
Baseline	88.2	67.1	5/15	33.3	89.8	71.4	5/15	33.3	88.7	68.4	5/15	33.3	77.3	46.8	4/15	26.7
Ours	**100.0**	**86.3**	**8/15**	**53.3**	**100.0**	67.2	5/15	33.3	**99.7**	**89.9**	**8/15**	**53.3**	**99.1**	**62.1**	4/15	26.7
Dependency Test Set																
Baseline	89.3	20.5	0/15	0.0	78.1	16.2	0/15	0.0	78.4	17.2	1/15	6.7	73.9	11.8	1/15	6.7
Ours	**99.3**	**96.9**	**12/15**	**80.0**	**95.2**	**80.4**	**7/15**	**46.7**	**93.3**	**84.4**	**10/15**	**66.7**	**98.6**	**81.7**	**7/15**	**46.7**

Table 2. Success rate (%) for grasp timing control under different temporal offsets.

Model	**Positive Offset (s)**			**Negative Offset (s)**		
	+0.5	+1.0	+1.5	-0.5	-1.0	-1.5
LINGO	0.0	0.0	0.0	0.0	0.0	0.0
Ours(SR)	84.7	78.0	76.0	88.0	75.3	37.3

Table 3. Statistical Results of timewarp model under different temporal offsets (Frame Shift)

Offset	**mean**	**std**	**25%**	**50%**	**75%**
-1.5	-1.260	0.404	-1.4	-1.4	-1.3
-1.0	-0.837	0.262	-1.0	-0.9	-0.9
-0.5	-0.434	0.186	-0.5	-0.5	-0.4
0.5	0.454	0.414	0.4	0.5	0.5
1.0	0.944	0.459	0.9	1.0	1.1
1.5	1.430	0.437	1.5	1.5	1.6

Table 3 reports the realised frame shift relative to each intended offset. The DTW alignment used for success-rate evaluation is also applied here, and motions exceeding a predefined DTW threshold are excluded. As shown in the table, the interquartile range (25%–75%) is narrow and consistent across all conditions, indicating stable variability in the refined trajectories. This suggests that the reduced success rate in the -1.5 s condition is primarily due to insufficient temporal shift rather than increased dispersion in trajectory outcomes. For preliminary estimation, we use a partial denoising step of $t = 30$, and a detailed analysis of partial-step behaviour is provided in the supplementary material.

6.3. Multi-Agent Scalability Evaluation

To evaluate the scalability and end-to-end robustness of **SyncMos**, we analyse chained *handover* interactions with varying numbers of agents across two scenes (*House* and *Restaurant*). Each motion sequence forms a linear dependency chain,

$$\text{agent}_1 \rightarrow \text{agent}_2 \rightarrow \cdots \rightarrow \text{agent}_N,$$

where each agent hands an object to the next. This setup naturally tests whether temporal errors accumulate and whether spatial handover quality deteriorates as the interaction chain becomes longer. We vary the number of agents

$$N \in \{2, 3, 5, 10\},$$

and run the full SyncMos pipeline, including dependency-aware event planning, spatial grounding, motion generation, and temporal refinement, to synthesise complete multi-agent handover sequences.

We report three metrics that jointly capture temporal and spatial stability:

- **Temporal Synchronisation Magnitude (TSM)**: number of frames each agent need to be shifted from its original best-contact moment for synchronisation.
- **Temporal Synchronisation Error (TSE)**: the deviation between the planned and realised handover frames for each pair of agents.
- **Contact Distance (CD)**: the minimum summed wrist-to-target distance at the interaction moment, indicating handover quality.

Figure 6 plots these metrics as a function of the number of agents. Across both scenes, TSM increases only mildly as N grows high, while TSE remains stable with no evidence of temporal error accumulation, even in 10-agent chains. This indicates that SyncMos prevents timing drift along longer dependency paths, and that the refinement step introduces only small, localised perturbations to the underlying motion. Meanwhile, Contact Distance remains within a reasonably stable range across high agent counts, confirming that spatial handover quality does not degrade as more agents participate. Minor differences between scenes are attributable to layout complexity, but the general trend remains consistent across environments.

These results demonstrate that SyncMos scales reliably to larger groups of agents: temporal alignment remains stable, dependency order is preserved, and handover quality is maintained even for 10-agent chains, highlighting SyncMos as an effective and lightweight post-generation synchronisation module. More test cases are in the supplementary material.

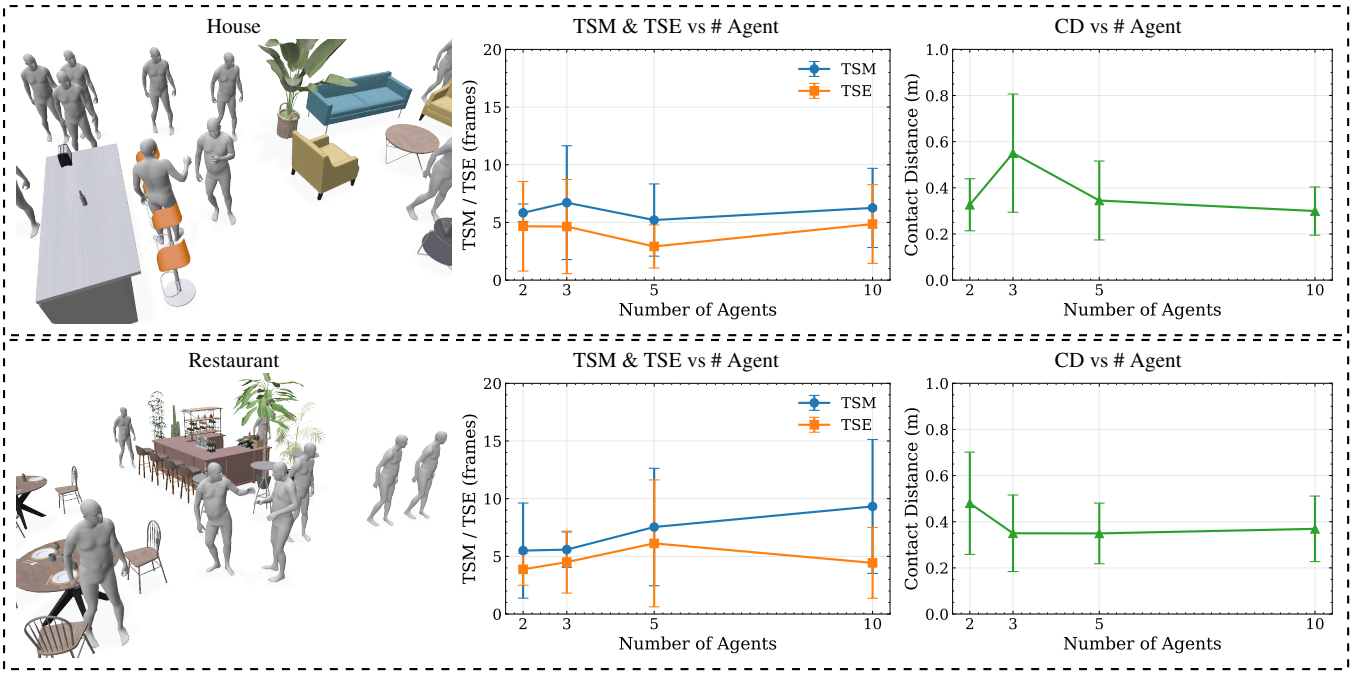

Figure 6. Multi-agent scalability across two scenes. We plot TSM, TSE, and CD metrics as the number of agents increases, with each row showing results from one scene.

7. Conclusion

We introduced SyncMos, a framework for generating temporally synchronized multi-agent motion using a single-agent diffusion model and an LLM-based dependency-aware planner. The planner extracts structured event sequences with sequential and parallel relations, and the proposed temporal synchronisation module aligns cross-agent timing through time-warping and diffusion-based refinement. Experiments on multi-agent scenarios demonstrate that SyncMos can enforce temporal dependencies and produce coordinated behaviors without requiring multi-agent model training.

Limitation and futurework Although SyncMos achieves consistent temporal alignment, several limitations remain. Large timing adjustments may reduce refinement stability, and LLM-based planning can introduce errors in long or ambiguous narratives. The current synchronisation module performs time-warping within a fixed motion duration and cannot modify the overall motion length. Extending the method to support temporal scaling of the motion itself remains an interesting direction for future work. The current 2D spatial grounding also limits fine-grained 3D contact reasoning. Moreover, because our current implementation relies on LINGO as the single-agent backbone, the generated motions are naturally constrained by the limitations of that model. These backbone-specific artifacts are independent of our synchronisation mechanism, and we expect SyncMos to benefit directly from future advances in single-agent diffusion-based motion generators. While the framework is model-agnostic, extending the evaluation to additional motion models is an important next step. Future work also includes incorporating physics- or contact-aware constraints, enabling tighter feedback between planning and generation, and expanding the system to more diverse or interactive settings.

Impact and value SyncMos introduces a novel formulation for synchronised multi-agent motion generation by coordinating multiple agents without requiring a multi-agent model. By combining single-agent generation with planning and temporal refinement, the method reduces model complexity and offers practical utility across applications. We expect SyncMos to provide a useful basis for future work on scalable scene-level interactions.

8. Acknowledgements

This work was supported by the Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism, South Korea (Project Number: RS-2024-00399136). This work was also supported by the Institute of Information and Communications Technology

Planning and Evaluation (IITP) grant (No. RS-2020-II201373).

References

[1] Joao Pedro Araújo, Jiaman Li, Karthik Vetrivel, Rishi Agarwal, Jiajun Wu, Deepak Gopinath, Alexander William Clegg, and Karen Liu. Circle: Capture in rich contextual environments. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 2

[2] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, 1994. 6

[3] Zhongang Cai, Jianping Jiang, Zhongfei Qing, Xinying Guo, Mingyuan Zhang, Zhengyu Lin, Haiyi Mei, Chen Wei, Ruisi Wang, Wanqi Yin, Liang Pan, Xiangyu Fan, Han Du, Peng Gao, Zhitao Yang, Yang Gao, Jiaqi Li, Tianxiang Ren, Yukun Wei, Xiaogang Wang, Chen Change Loy, Lei Yang, and Ziwei Liu. Digital life project: Autonomous 3d characters with social intelligence. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 2

[4] Zhi Cen, Huaijin Pi, Sida Peng, Zehong Shen, Minghui Yang, Zhu Shuai, Hujun Bao, and Xiaowei Zhou. Generating human motion in 3d scenes from text descriptions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 1

[5] Jianqi Chen, Panwen Hu, Xiaojun Chang, Zhenwei Shi, Michael Kampffmeyer, and Xiaodan Liang. Sitcom-crafter: A plot-driven human motion generation system in 3d scenes. In *Int. Conf. Learn. Represent.*, 2025. 1, 2

[6] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *Int. Conf. Learn. Represent.*, 2023. 2

[7] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *ACM Int. Conf. Multimedia*, 2020. 2

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Adv. Neural Inform. Process. Syst.*, 2020. 5

[9] Nan Jiang, Zimo He, Zi Wang, Hongjie Li, Yixin Chen, Siyuan Huang, and Yixin Zhu. Autonomous character-scene interaction synthesis from text instruction. In *SIGGRAPH Asia Conference Papers*, 2024. 1, 2, 4, 5

[10] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Int. Conf. Comput. Vis.*, 2023. 2

[11] Manmyung Kim, Kyunglyul Hyun, Jongmin Kim, and Jehee Lee. Synchronized multi-character motion editing. *ACM Trans. Graph.*, 2009. 2

[12] Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-human motion generation under complex interactions. *Int. J. Comput. Vis.*, 2024. 1, 2

[13] Donggeun Lim, Jinseok Bae, Inwoo Hwang, Seungmin Lee, Hwanhee Lee, and Young Min Kim. Event-driven storytelling with multiple lifelike humans in a 3d scene. In *Int. Conf. Comput. Vis.*, 2025. 1, 2, 3, 4, 6

[14] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Adv. Neural Inform. Process. Syst.*, 2023. 2

[15] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[16] Omid Taheri, Vasileios Choutas, Michael J Black, and Dimitrios Tzionas. Goal: Generating 4d whole-body motion for hand-object grasping. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[17] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *Int. Conf. Learn. Represent.*, 2023. 2

[18] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[19] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Adv. Neural Inform. Process. Syst.*, 2022. 3

[20] Andrew Witkin and Zoran Popovic. Motion warping. In *SIGGRAPH Conference Papers*, 1995. 2, 5

[21] Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. Motionstreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. In *Int. Conf. Comput. Vis.*, 2025. 2

[22] Hongwei Yi, Justus Thies, Michael J. Black, Xue Bin Peng, and Davis Rempe. Generating human interaction motions in scenes with text control. In *Eur. Conf. Comput. Vis.*, 2024. 1

[23] Yuhong Zhang, Jing Lin, Ailing Zeng, Guanlin Wu, Shunlin Lu, Yurong Fu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x++: A large-scale multimodal 3d whole-body human motion dataset. *arXiv preprint arXiv:2501.05098*, 2025. 2