Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit Nr. 3624946

Multimodal LLM for Theory of Mind Modeling in Collaborative Tasks

Jan-Philipp Thewes

Course of Study: Informatik

Examiner: Prof. Dr. Andreas Bulling

Supervisor: Matteo Bortoletto, M.Sc.

Commenced: December 11, 2023

Completed: June 11, 2024

CR-Classification: I.7.2

Abstract

The ability to infer the beliefs, desires, and intentions of others, known as Theory of Mind (ToM), is crucial for effective collaboration. In this work, we explore this ability in the context of task-oriented human-machine collaboration with a focus on Multimodal Large Language Models (MM-LLMs). While previous works relied on fixed question-answer pairs or explicit ToM modeling, we investigate the implicit ToM capabilities of MM-LLMs within the multimodal research environment *Mindcraft*. We propose a model architecture that integrates video, text, and knowledge graphs to create a more realistic and flexible collaborative interface. Our findings show that MM-LLMs not only outperform specialized baseline models in ToM tasks but also achieve human performance in some scenarios. Furthermore, our model accurately predicts its own and the partner's missing knowledge in collaborative situations, demonstrating its potential for common-ground reasoning. However, the importance of multimodality for ToM tasks could not be confirmed in our experiments, suggesting that task-specific video sampling and encoding might be crucial for successful multimodal reasoning. Overall, this work reinforces the potential of MM-LLMs to enable more intuitive and efficient human-machine collaborations while surpassing previous baselines in ToM task performance within multimodal environments.

Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

AI Artificial Intelligence. 17

CE Cross-Entropy. 29

CPA Collaborative Plan Acquisition. 13, 33

LLM Large Language Model. 11

LoRA Low-Rank Adaptation. 19

LSTM Long Short-Term Memory. 24

MHA Multi-Head Attention. 16

MLP Multi-Layer Perceptron. 16

MM-LLM Multimodal Large Language Model. 11

MToM Machine Theory of Mind. 11, 14

PEFT Parameter-Efficient Fine-Tuning. 19

QLoRA Quantized Low-Rank Adaptation. 19

ToM Theory of Mind. 11

1 Introduction

Whenever humans interact with each other they intuitively build a mental model about the other person. This allows humans to adapt to collaborative situations very effectively, as this mental model helps to predict the other agents' goals, beliefs and actions. This ability to model other agents' mental states is referred to as Theory of Mind (ToM) in behavioral psychology [PW78]. ToM allows us to reason about others' behavior as a whole, even with little or incomplete verbal interaction. This modeling of agent beliefs happens disconnected from the actual underlying mechanism which builds those beliefs in the other agents but happens in a rather abstract manner [GW92]. It helps humans subconsciously in various daily coordination tasks, such as navigating subways, avoiding collisions, or passing through doors. As an example, imagine driving or walking in traffic. You anticipate the actions of other drivers or pedestrians based on your understanding of their likely intentions. This helps in making safe decisions and avoiding accidents.

Moreover, ToM plays a crucial role in more formal collaborations between humans, making it essential to explore this ability also in the context of human-machine collaboration. While humans learn this ability between the age of two and five years in a progressive sequence [CKH13], it is a matter of current research if and to what extent artificial intelligences possess this ability [MLZ+23; ZZW24].

With the emergence of deep neural networks, artificial agents are increasingly interacting with humans in collaborative tasks. Machine Theory of Mind (MToM) has become a target here, aiming to enable artificial agents to model mental states for seamless collaboration [MLZ+23; RPS+18].

At the same time, Large Language Models (LLMs) are becoming increasingly successful and popular in conversational interactions with humans [CLL+23; TGZ+23; TLI+23; TMS+23]. However, most of the popular and powerful LLMs are limited to text as the major input modality. In contrast, real-life situations are inherently multimodal, which leaves already existing solutions limited in their information and understanding capabilities. Consequently, research into the field of Multimodal Large Language Models (MM-LLMs) has grown recently [ADL+22; LHW+23; LLSH23; WFQ+23; ZLB23].

In contrast to already existing research on Multimodal LLMs, we focus on their use in collaborative settings with analysis of Machine Theory of Mind. Therefore, we propose the use of MM-LLMs for the use in collaborative tasks. We evaluate the performance of our multimodal architecture in Theory of Mind and Collaborative Plan Acquisition tasks within a 3D game environment called *Mindcraft* [BCC21; BMY+23]. The model architecture is based on the work of Wu et al. [2023] and adapted towards the use in task-oriented collaboration within the Mindcraft environment.

Previously, research in ToM focused on explicitly modeling an architecture that supports ToM in a neural network [NNL+23; RPS+18] or predicting only on trained question-answer pairs [BCC21]. As recent research found out though, LLMs seem to have developed the ability to model the belief status of agents without explicit architectural modeling [BRSB24; ZZW24]. Therefore, we do not make use of explicit ToM modeling in our architectures but test the implicit ToM capabilities of LLMs.

Inherently with LLMs, we are not limited to a fixed question-answer prediction but can benefit from the learned reasoning and natural language capabilities. Since in collaborative settings with real humans fixed question-answer pairs limit the interaction dramatically, LLMs seem to be a good fit for more intuitive interactions. The ultimate aim is to enhance collaboration through the use of multiple modalities and generalized models, moving beyond fixed question-answer predictions and promoting more seamless human-machine interactions.

Within our work we aim to answer the following research questions:

RQ1: Are LLMs capable of performing Theory of Mind tasks in situated dialogues?

RQ2: Do Multimodal LLMs outperform specialized models targeting ToM tasks?

RQ3: How important is multimodality for ToM tasks?

RQ4: Can Multimodal LLMs effectively predict missing knowledge in collaborative situations?

Our work shows that Multimodal LLMs can be used for ToM tasks and that they outperform the smaller but more specialized baseline models. In some tasks, our model even reaches human performance. This implies that LLMs possess Machine Theory of Mind abilities to the limited extent tested here. However, the importance of multimodality for ToM tasks could not be confirmed in our experiments. This is assumed to be related to the video sampling and training data in our experiments. With better-suited video encoding and data, multimodality could be more important. Finally, our model is able to predict missing knowledge of agents in collaborative situations with high accuracy if fine-tuned on such tasks. Therefore, the use of Multimodal LLMs for human-machine collaboration is confirmed to be a promising path.

In summary, our contributions are the following:

(i) We design an approach for targeting MM-LLMs towards Theory of Mind within the 3D game environment *Mindcraft*. Within this approach, we propose a new multimodal architecture targeted towards harnessing the pre-trained capabilities of LLMs in collaborative settings.

(ii) We evaluate the performance of our approach in the Mindcraft environment and compare it to previous baselines on these ToM tasks [BCC21; BMY+23; BRA+24]. With this, we set a first baseline of MM-LLMs in Theory of Mind tasks.

(iii) We evaluate our model furthermore on tasks of Collaborative Plan Acquisition (CPA) to explore the performance of our model in this important step towards a common ground.

2 Related Work

This work combines the fields of Machine Theory of Mind, Multimodal Large Language Models, and human-AI collaboration by testing the ToM capabilities of MM-LLMs in the context of multi-agent collaboration with the *Mindcraft* dataset.

2.1 Machine Theory of Mind

Theory of Mind is a fundamental aspect of human social cognition, enabling us to understand and predict the thoughts and intentions of others. The core of ToM is the ability to attribute mental states to others, such as beliefs, desires, and intentions [HS44; PW78; YDF08]. The origins of the concept of Theory of Mind can therefore be traced back to seminal works in human psychology and cognitive science. The term itself was first introduced in 1978 by Premack and Woodruff in their paper, "Does the chimpanzee have a theory of mind?" [PW78]. Even earlier, Heider and Simmel's 1944 study on apparent behavior provided insights into the attribution of intentionality to moving shapes, suggesting that humans naturally ascribe intentions and purposes to even simple shapes [HS44].

Recently it has been explored, whether not only humans or chimpanzees possess this ToM but also intelligent machines can attribute mental states to other agents, therefore enhancing their collaboration capabilities. For instance, the work by Rabinowitz et al. [RPS+18] introduced the concept of Machine Theory of Mind (MToM) while differing from previous works on Bayesian ToM approaches [BJST17; BS11] in a way that agents need to autonomously model other agents with limited data. Rabinowitz et al. formulated MToM as a meta-learning problem, where an agent learns to model other agents through observation from a third-person perspective. For this task, they proposed the *Theory of Mind Network* (ToMnet) based on deep learning.

After Rabinowitz et al.'s foundational work on Machine ToM, several lines of research emerged [BSB24; NBM19; NNL+22; OCSS23; OHS21]. Some focus on explicitly crafting model architectures that foster the development of further levels of ToM [NNL+22; SKW+23], while others focus on using ToM modeling to support explainability [ALS+19; OHS21]. There also exist various approaches, which type of machine learning is best

for fostering ToM in intelligent systems. Bayesian approaches [NBM19; WWE+20; ZGP+22] as well as incorporating ToM into the reward function of reinforcement learning [OCSS23] showed promising results. However, recently much research has gone into the ToM abilities of Large Language Models.

While some point out that Large Language Models lack robust Theory of Mind abilities [SLFC23; Ull23], others have found evidence for their emergence in the most recent LLMs [LCS+23; ZZW24]. For example, Zhu, Zhang, and Wang lately found that LLMs have internal representations as part of their activations for the mental states of other agents and that manipulations of those representations strongly influence their social reasoning performance [ZZW24]. In opposition, Ullman [2023] argues against the ToM abilities of LLMs, as he found that LLMs perform badly on slight variations of classical ToM tests, while also noting that ToM tests used for children might not be valid for LLMs or machines in general [Blo81]. Other works explored how ToM abilities can be best supported through prompting techniques or explicit modeling [LTV23; MH23; SKW+23]. For example, Moghaddam and Honey [2023] found that prompting techniques such as step-by-step thinking, few-shot learning, and chain-of-thought reasoning [BMR+20; KGR+23; WWS+23] can improve the ToM abilities of LLMs significantly [MH23]. These contradicting findings suggest that more research and standardized evaluations are needed in this field.

However, in contrast to fields that have a defined task specification and a clear framework for resolution, Machine ToM does not have a standardized benchmark dataset, complicating the formal comparison of different approaches [MLZ+23]. While efforts for standardization in some categories exist, e.g. for ToM in LLMs [CWZ+24; KSZ+23; MSPC23], an overarching cross-architectural evaluation is still difficult.

Our work aims to contribute to the field of Machine Theory of Mind by testing the ToM capabilities of LLMs in an existing multimodal benchmark dataset. For that, we take inspiration from the mentioned techniques such as few-shot prompting while providing more indications towards the overarching question about the existence of ToM in LLMs.

2.2 Multimodal LLMs

The transformer architecture, which is the core building block for LLMs, was introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017 [VSP+23]. While there are several variants for transformer-based models, we focus on the use of autoregressive decoder-only transformer models [ENO+21]. In an autoregressive transformer model $M : X \to Y$, an input sequence of tokens $x = [x_1, \ldots, x_t]$ from a vocabulary is processed

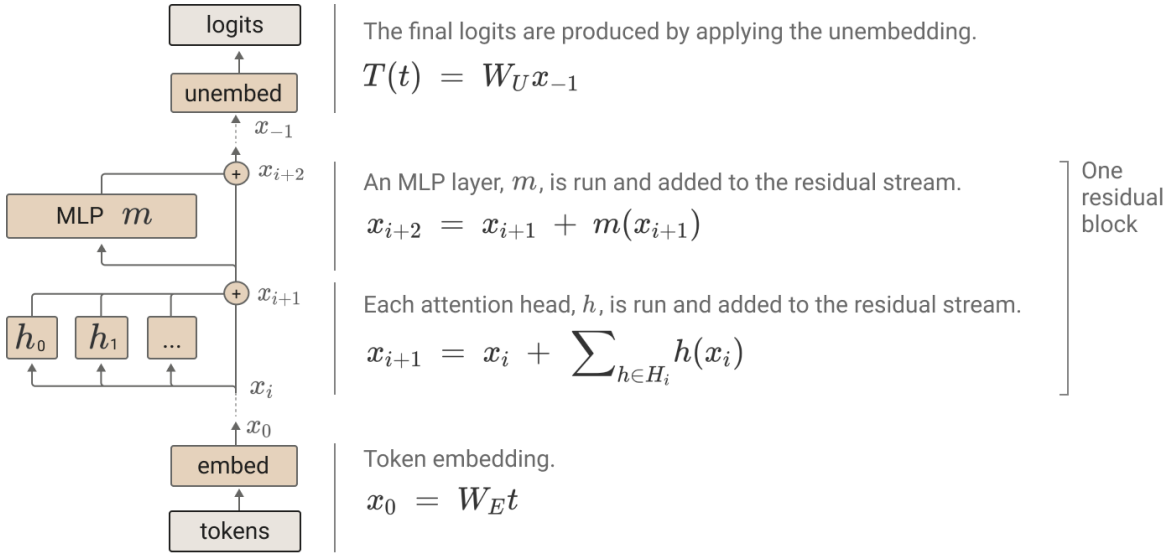

Figure 2.1: Transformer Language Model Architecture [ENO+21]

to predict the next token by producing a probability distribution over the vocabulary V, the so-called logits [GNP+23]. The tokens first get embedded before the model proceeds with a series of residual blocks. These residual blocks consist of Multi-Head Attention (MHA) and Multi-Layer Perceptron (MLP) layers, both of which interact with a central residual stream through linear projections (see Figure 2.1). The residual stream is the sum of outputs from all previous layers. The MHA layers facilitate information transfer across token positions while the MLP layers primarily handle feature extraction [ENO+21; GNP+23].

Using this architecture, popular Large Language Models (LLMs) like OpenAI's GPT-4 [Ope23] and others show advanced language understanding and reasoning, thanks to additional techniques such as instruction tuning [OWJ+22] and reinforcement learning from human feedback (RLHF) [SOW+22]. The development of open-source LLMs such as Flan-T5 [CHL+22], Vicuna [CLL+23], Mistral [JSM+23], LLaMA [TLI+23], and Alpaca [TGZ+23] made it possible, that the wider community could base their research upon those. This laid the groundwork for the creation of Multimodal Large Language Models (MM-LLMs) that handle diverse inputs and tasks.

The argument for MM-LLMs is threefold. Firstly, they align more closely with human perception, which naturally integrates multisensory inputs that are often complementary. This multimodal integration is expected to enhance the intelligence of MM-LLMs. Secondly, MM-LLMs offer a more user-friendly interface, allowing for more flexible interactions and communication with intelligent systems. Lastly, MM-LLMs are more versatile task solvers, supporting a broader spectrum of tasks beyond the capabilities of traditional LLMs [YFZ+23].

Researchers have been developing MM-LLMs by integrating encoders from various modalities with the textual capabilities of LLMs, enabling these models to process non-textual inputs. For instance, Flamingo employed a cross-attention mechanism to feed image encodings into LLMs [ADL+22], while BLIP-2 used a Q-Former to interpret image queries for LLMs [LLSH23]. LLaVA projected image features directly into the word embedding space [LLWL23]. Similar approaches have been applied to create MM-LLMs that comprehend video (Video-Chat, Video-LLaMA) and audio (SpeechGPT) [LHW+23; ZLB23; ZLZ+23]. Notably, PandaGPT extended this integration to six different modalities using the ImageBind multimodal encoder [SLL+23]. NExT-GPT went a step further and also integrated multimodal diffusion decoders to the output side of LLMs and therefore achieved multimodal output feeds [WFQ+23].

In the context of ToM in multimodal settings, Jin et al. took a different approach to enable their model to perceive multiple modalities by introducing BIP-ALM (Bayesian Inverse Planning Accelerated by Language Models). BIP-ALM works by transforming the different modalities into symbolic state representations, using LLMs to generate the representations from the text input, and then using an Inverse Symbolic Planner for prediction. This work showed to be more robust in answering ToM questions than common MM-LLMs [JWC+23].

Within this field of multimodal LLMs, our model strongly relates to the approaches of NExT-GPT, PandaGPT and Video-Chat by using pre-trained encoders to integrate video input into the LLM's embedding layer [LHW+23; SLL+23; WFQ+23]. In contrast to BIP-ALM, we use the capabilities of the pre-trained LLM as the core module for answering ToM questions.

2.3 Human-AI Collaboration and Mindcraft

Recent advances in Artificial Intelligence (AI), particularly in the domain of Theory of Mind, have enhanced the potential for effective collaboration between artificial agents and humans. As mentioned, the concept of ToM is crucial in understanding and predicting the behavior of collaborative partners [KLJ+07; LCS+23]. This understanding is also relevant in the context of task-oriented collaboration. In contrast to other forms of collaboration, task-oriented collaboration involves the completion of a shared, specific and often pre-defined goal. Other forms of collaboration can have more open-ended objectives, such as the exchange of information or the development of a shared understanding [LTAE24]

Testing the ability of effective task-oriented collaboration has been of interest in research for years. Early approaches in this domain focused on assessing common ground in

dialogues, as highlighted by Udagawa and Aizawa in their 2019 study [UA19]. They explored the dynamics of dialogues in collaborative tasks, emphasizing the role of mutual understanding in effective communication. Another early area of exploration was physical world evaluations of task demonstrations [MFS+02]. This approach involved analyzing how AI systems and humans perform and interpret physical tasks in a shared environment, offering insights into the nuances of collaborative actions.

Recently, there has been a trend towards utilizing game environments as platforms for studying human-AI collaboration. This shift is well-represented in works like those of Suhr et al. [SYS+22], Narayan-Chen et al. [NJH19], and Jayannavar et al. [JNH20]. These studies leveraged the controlled yet complex scenarios offered by 3D environments to simulate and analyze collaborative interactions, providing a rich ground for understanding the dynamics of human-AI cooperation.

However, also text-based game environments are utilized, as in the work of Li et al. from 2023. They observed that LLM-based agents demonstrated emergent collaborative behaviors and high-order Theory of Mind capabilities, although they faced challenges with long contexts and hallucinations [LCS+23].

Among these contributions, the work of Bara, CH-Wang, and Chai (2021) stands out as particularly relevant to our research. They introduced a new approach by integrating ToM modeling into the context of human-AI collaboration through the use of the 3D virtual blocks world of Minecraft. Their study involved the creation of a dataset based on collaborative tasks performed by human pairs within this environment. Building on this, Bara, CH-Wang, and Chai developed task definitions within their customized game, *Mindcraft*, specifically designed to explore human-AI collaboration in relation to ToM [BCC21]. Their paper emphasized the importance of ToM in maintaining common ground during human-AI collaboration while demonstrating how collaborative partners with asymmetric knowledge and skills can achieve joint goals. Furthermore, it showed how their beliefs of each other evolve and converge over time [BCC21].

We will use this experimental setup *Mindcraft* in our work to test the ToM capabilities of MM-LLMs in task-oriented collaboration. Therefore, our work can be seen as an extension of their research into the domain of multimodal LLMs.

2.4 Training LLMs

Many of the openly available LLMs are general-purpose LLMs, sometimes instruction-tuned to work better in a chat-like setting. For adapting those LLMs to the specific task to be solved, one can either use prompt engineering to tweak the model output through

in-context learning [DLD+23] or fine-tune the model parameters for the downstream task [HSW+21]

It has been shown that using specific prompt formats can have a significant influence on the model performance and that LLMs are fairly sensitive to small changes in the prompt [SCTS23; WFH+23]. Moghaddam and Honey also showed that performance in ToM questions improved when prompting techniques such as Step-by-Step Thinking or Few-Shot Chains were used [MH23].

While prompt engineering has the advantage of not needing to train the model, it has its limitations when it comes to more specific tasks. Therefore, recent research has also been focusing on efficient and performant fine-tuning techniques. In most cases, one wants to keep the pre-trained model's reasoning logic but wants to adjust the output behavior, while not using computational resources similar to full training. For that, fine-tuning techniques such as adapter-based Parameter-Efficient Fine-Tuning (PEFT) and Low-Rank Adaptation (LoRA) gained popularity [HSW+21; HWL+23]. While there are many more efficient fine-tuning techniques available [XXQ+23], the focus of our work will be on using LoRA.

When applying the LoRA technique, the pre-trained model weights are kept frozen and trainable rank decomposition matrices are injected into each layer of the Transformer architecture (see Figure 2.2) [HSW+21]. This reduces the number of trainable parameters drastically. Doing so, LoRA achieves a higher training throughput, no additional inference latency and a lower memory footprint, while performing on par compared to full fine-tuning [HSW+21].

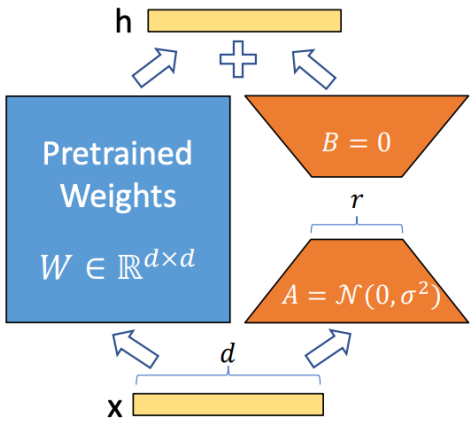

Figure 2.2: LoRA fine-tuning: Only A and B is trained [HSW+21]

Closely related to fine-tuning techniques, lies the challenge of limited GPU VRAM of common graphic cards when fine-tuning LLMs. During training, the model parameters and additionally up to 8 bytes per parameter for the optimizer state have to fit into the GPU VRAM [KB17]. While LoRA already reduces the footprint by reducing the number of trainable parameters, the parameters of the base model still have to fit onto the GPU in full precision of 16 bits. This can already bear a challenge as already smaller LLMs have 7B parameters. Quantized Low-Rank Adaptation (QLoRA) addresses this challenge by using 4-bit quantization for the model parameters, which reduces the memory footprint by 2-3x [DPHZ23]. Other techniques for reducing the memory cost are gradient checkpointing and 1-bit LLMs [CXZG16;

MWM+24]. Instead of storing all activations, in gradient checkpointing only a subset of the activations are stored while the others are recomputed during backpropagation [CXZG16]. On the other side, the recent idea of 1-bit Transformers reduces the memory required by representing the model parameters not as 16-bit Floats but as ternary values out of {-1, 0, 1} [MWM+24].

In addition to the dependency on parameter count, the memory and compute requirements grow quadratically with the number of tokens in the input [KWH22]. This is inherent to the original attention mechanism, but techniques such as *Flash Attention* [DFE+22] and *Dilated Attention* [DMD+23] have been introduced to limit this influence by changing the attention mechanism used. While Flash Attention reduces the number of memory reads and writes between GPU memory hierarchies through tiling [DFE+22], Dilated Attention achieves linear complexity by using sparsified and exponentially decreasing attention allocations [DMD+23]. However, these techniques are not yet widely used for the commonly available LLMs.

Considering those challenges, the engineering effort while fine-tuning LLMs should not be neglected. Within our work, we will be exploring the benefits of in-context learning through few-shot prompting in ToM tasks. Additionally, we will fine-tune our model using QLoRA and gradient checkpointing to limit our computational consumption.

3 Method

3.1 Mindcraft

Our main interest in this work lies in the evaluation of Theory of Mind capabilities of MM-LLMs in collaborative settings. As our research aims towards human-machine collaboration, we want to evaluate the model's abilities in a setting that is as close to human-human collaboration as possible. At the same time, it should focus on task-oriented collaboration and have multimodal inputs to align with human perception. To combine these requirements, we selected the *Mindcraft* dataset and environment as the basis for our training and evaluation [BCC21; BMY+23].

The *Mindcraft* environment is a multimodal dataset and environment for testing ToM [BCC21]. It is based on the 3D virtual blocks game Minecraft, which was adapted for this purpose of collecting the *Mindcraft* dataset. This dataset was collected with the purpose of facilitating research on Theory of Mind in situated collaborative tasks [BCC21]. Moreover, it provides multimodal training data with video, text and graph information available. The overarching task within the dataset is to craft a joint target material by collaborating with another agent.

Therefore, for collecting the dataset pairs of human players were asked to jointly craft a shared target block. However, each player only had incomplete knowledge about how to craft this target block by means of combining other materials. This is reflected in an incomplete knowledge graph, which is shown to the agents (see Figure 3.1). Moreover, the players had different tools available to them, which could not be shared but were needed for specific steps of the crafting process. By having this information and skill asymmetry, the players had to communicate over the game chat and collaborate to achieve the joint goal. The players could see their own video feed, the game chat and their own knowledge graph.

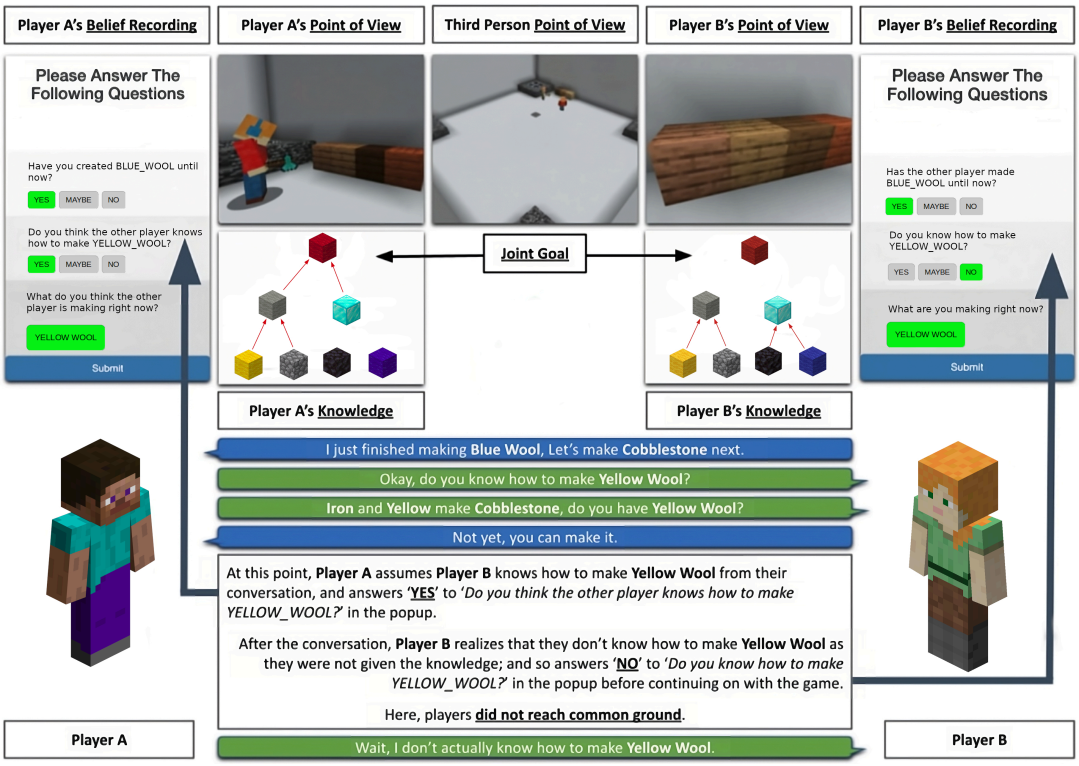

Figure 3.1: Example interaction in the Mindcraft environment, based on [BCC21]. The players have to interact through the game chat to achieve the joint goal of crafting a target material. Each player has an incomplete knowledge graph. The belief status of the players is probed with questions about their own and the other player's mental state.

ToM Tasks Every 75 seconds, the players were probed with three questions about their own or the other agent's mental state. Always three types of questions were asked:

- *Task Status*: Have you created Material A until now? / Has the other player made Material A until now?
- *Task Knowledge*: Do you know how to make Material A? / Do you think the other player knows how to make Material A?
- *Task Intention*: What are you making right now? / What do you think the other player is making right now?

These set of questions were asked in a way, that one player would always provide the ground truth answer for the other player's question (see Figure 3.2). With this setup, Bara, CH-Wang, and Chai collected the Mindcraft dataset consisting of the video frames, game chat, knowledge graph and question-answer pairs.

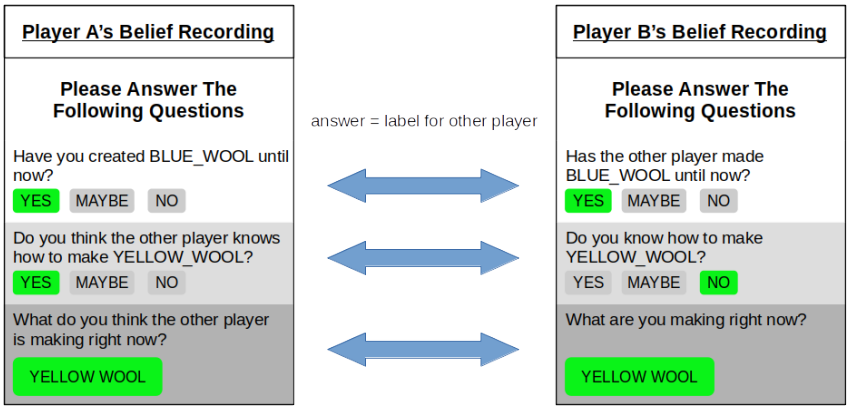

Figure 3.2: Example question set from the Mindcraft dataset [BCC21], representing the ToM tasks. The questions are paired to provide the ground truth answer for the other player's question.

When now evaluating the performance of a model in the Mindcraft environment, the model takes the perspective of one of the players and is probed with the same set of questions. Their accuracy in answering these questions is evaluated against the recorded ground truth and compared to the baseline performance. The three types of questions above are coined as *Task Status*, *Task Knowledge* and *Task Intention* questions and are referred to as *ToM tasks* in the context of this work.

Collaborative Plan Acquisition In addition to those ToM tasks, Bara et al. extended upon their work in 2023 and introduced a new target called *Collaborative Plan Acquisition* (CPA). Because collaborative agents with incomplete knowledge suffer from an incomplete action space if they do not know the complete plan, acquiring a complete plan jointly is stated to be a crucial step towards effective collaboration. "It's therefore important for an agent to predict what knowledge is missing for themselves and for their partners, and proactively seek/share that information so the team can reach a common and complete plan for the joint goal" [BMY+23]. Therefore the following problem to be solved was introduced:

Definition 3.1.1 (Collaborative Plan Acquisition Problem)
"In a collaborative plan acquisition problem with a joint plan P, an agent i and its collaborative partner j start with partial plans $P_i = (V, E_i)$ *and* $P_j = (V, E_j)$*. [..] The problem is for agent i to acquire its own missing knowledge* $E_i = E \setminus E_i$ *and the partner j's missing knowledge* $E_j = E \setminus E_j$*" [BMY+23]. The joint and partial plans are represented as a graph* $P = (V, E)$ *with vertices* V *and edges* E*.*

Two tasks were introduced to test this ability of acquiring a joint plan.

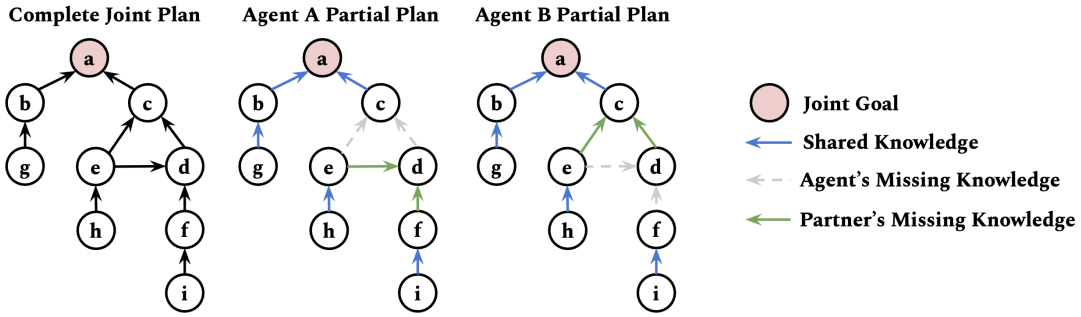

Figure 3.3: Schematic knowledge graphs visualizing the agents' missing knowledge for the CPA tasks [BMY+23]. Within the CPA tasks, the agents need to predict the missing edges in their own and the partner's plan.

CPA Task 1: Inferring Own Missing Knowledge. The agent needs to predict the edges that are missing in his plan in comparison to the complete joint plan. For example in Figure 3.3 agent A needs to infer the edges $d \rightarrow c$ and $e \rightarrow c$ in his plan.

CPA Task 2: Inferring Partner's Missing Knowledge. The agent infers the missing edges in the partner's plan in comparison to the complete joint plan. In the example in Figure 3.3 agent A should predict the edges $e \rightarrow d$ and $f \rightarrow d$ in the partner's plan.

The CPA tasks are evaluated at the end of a game so that all information is available to the agent. Within this work, we use both the ToM and CPA tasks for a comparison of our approach to the original Mindcraft model architecture. The main focus of this work however lies on the ToM tasks, with the CPA tasks being an extension and further evaluation of the model's capabilities.

3.2 Architecture

While the original Mindcraft model architecture was based on Long Short-Term Memory (LSTM) [HS97] or simple Transformers [VSP+23], we adapted the architecture of Wu et al. [2023] towards the Mindcraft dataset so that an LLM model is the basis of the architecture. Therefore, our architecture does not use specific prediction heads for each question but uses the natural language output of the LLM as the prediction. This allows the architecture to be more flexible to changing tasks and to be used in a more general setting. Moreover, the architecture intentionally does not focus on explicitly modeling ToM capabilities but on the multimodal capabilities of the model.

The multimodal model architecture of NExT-GPT was taken as inspiration due to its versatility and extendability [WFQ+23]. In addition, the training regime was consistent

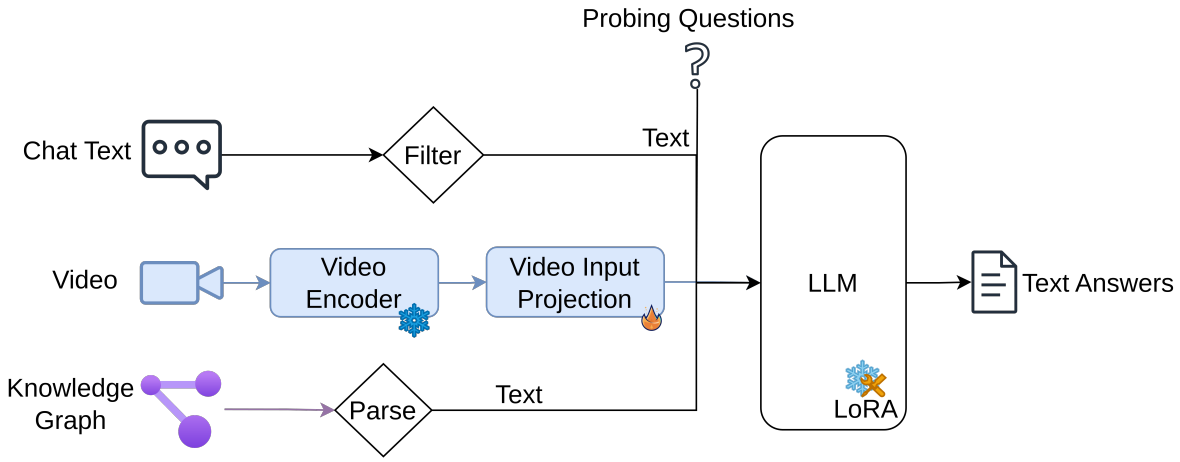

Figure 3.4: Our multimodal model architecture. Chat text, knowledge graph and probing questions are passed as text to the LLM. The video frames are encoded using a pre-trained video encoder and projected into the LLM embedding space. The LLM can be kept frozen or fine-tuned using LoRA, the projection layer is fully trained.

with our intended approach. With this chosen architectural approach, future extensions with more modalities as input or output are feasible to enable more intuitive interfaces for collaboration.

For our model architecture the main input modalities are the video frames, the game chat and the knowledge graph (see Figure 3.4). We pass the game chat to the LLM as text input after being filtered to only contain messages that are considered relevant for the task. Which chat messages are considered relevant is determined in a preprocessing step by looking at the so-called *dialogue moves* associated with them. Dialogue moves are labels given in the dataset to each chat message, which indicate the type of information the message contains (see Figure A.1). Exemplary dialogue move labels are "Statement-Recipe", "Statement-Goal" or "Directive-Make". We selected a subset of the dialogue moves that indicate meaningful information for the ToM and CPA tasks and considered only chat messages with these labels for the LLM input (see Listing A.2). We used this filtering approach to reduce the memory requirements during fine-tuning, which grow quadratically with the number of tokens (see Section 2.4).

As the second input modality, we use the video frames from the game. These are encoded using a pre-trained video encoder and then projected into the embedding space of the LLM using a linear layer. The video encoder is kept frozen, while the projection layer is trained together with the rest of the model. Leaning onto the work of Wu et al. [2023], we use ImageBind as the video encoder in this work [GEL+23].

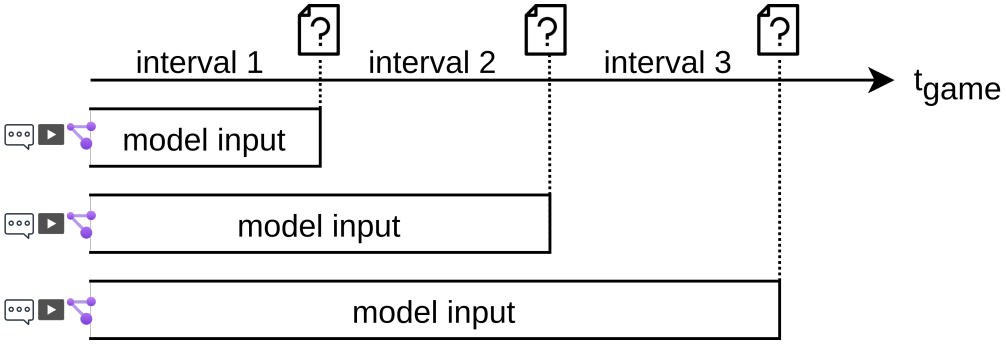

Figure 3.5: Schema for passing the multimodal input to the model. For each question at the end of a game interval, the model receives all the information available until that time.

The third input modality is the knowledge graph, which we pass to the LLM as text input after parsing it into a standardized format. Finally, the probing questions are passed to the LLM as text input as well.

We use a pre-trained and instruction-tuned Mistral model with 7 billion parameters (Mistral-7B-Instruct-v0.2 [1]) [JSM+23] as the core LLM of our architecture. We selected Mistral 7B for its balance between performance and model size, while still being publicly available. Depending on the training regime, the LLM is either fine-tuned on the Mindcraft dataset using QLoRA [DPHZ23; HSW+21] or kept frozen (see Table 3.1).

As mentioned in Section 3.1, for the evaluation of the ToM tasks the players get asked three questions every 75 seconds. Therefore, the games are split into intervals of 75 seconds and the model is probed with the questions at the end of each interval. When the model is prompted with the questions referring to e.g. interval three in a game, it needs to know all information prior to this timestamp. Hence, we pass all the chat messages and video frames until this time to the model. The parsed information from the knowledge graph is constant throughout a game. This results in the input passing scheme depicted in Figure 3.5, which shows that the model input size grows for intervals further into the game. With this, the model does not have to hold memory of previous intervals, as the baseline LSTM model did, but can rely on the input passing scheme to have all information available at the time of the question.

[1] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

3.3 Prompt Design

We structured the textual prompt format in five parts (see Listing 3.1). First, the system prompt gives some general context of the situation. Then the filtered chat messages are passed to the model, followed by the textual representation of the knowledge graph. The probing questions are then stated with the allowed answer options. Finally, we instruct the model on how to answer the questions.

```
You collaborate with a human in a 3D game like Minecraft, aiming to create a goal
material. Both of you lack knowledge and must exchange information to reach your goal.

Dialogue:
Human: i have an golden axe
You: i have diamond and; gray wool + golden axe
Human: emerald bloc = diamond block + blue wool

Own Material Knowledge:
IRON BLOCK + LAPIS BLOCK with GOLDEN AXE = SOUL SAND
CYAN WOOL + GRAY WOOL with GOLDEN AXE = IRON BLOCK
...
OAK PLANKS with IRON AXE = DIAMOND BLOCK

Questions:
Have you created GRAY WOOL until now? Options: NO,MAYBE,YES
Do you think the other player knows how to make SOUL SAND? Options: NO,MAYBE,YES
What do you think the other player is making right now? Take options from own material
knowledge. Additionally NOT SURE is an option.

Do not explain your answer. Restrict each answer to the three questions to one word
and separate them by a comma
```

Listing 3.1: Prompt example for ToM tasks. The video embedding is left out for clarity.

The prompt is designed to ask all three types of questions in a single prompt, to allow for a more efficient training process. Listing 3.1 shows an example prompt for the ToM tasks without video input. In settings where we also pass the video frames to the model, the video frames are inserted after the system prompt and enclosed by "*VIDEO: <VID> <FRAME EMBEDDINGS> </VID>*". The use of those special tokens is inspired by the work of Wu et al. [2023]. This joining between the text tokens from the prompt and the video embeddings happens in the embedding space of the LLM. Therefore, the prompt shown in Listing 3.1 would be first embedded using the LLMs embedding layer before combining it with the output of the video projection layer.

Setup	Modalities	Components	Training
Setup 0	Text	LLM	LLM inference only
Setup 1	Video, Text	Video Encoder, Projection Layer, LLM	Projection Layer only
Setup 2	Text	LLM	LLM QLoRA fine-tuning
Setup 3	Video, Text	Video Encoder, Projection Layer, LLM	Projection Layer full training, LLM QLoRA fine-tuning

Table 3.1: Properties of the different model setups used in this work. Comparison regarding their modalities and components being included or trained.

3.4 Implementation Details

Model Setups We aimed to explore the influence of the included modalities and trained components on the performance in the ToM tasks. Therefore, we used the model in different variants throughout our work and denoted them as separate *setups*. These *setups* are shown in Table 3.1. The included model components and modalities for each setup refer to these depicted in Figure 3.4, where the text modality contains the parsed knowledge graph. We selected this collection of setups as the most meaningful combinations of modalities and model components.

Memory Constraints With the architecture and input passing scheme described above, we treated each game interval as an independent training instance, containing all the information available at the time of the questions. This allowed us to make use of shuffling within the training set to mitigate the influence of the order of the intervals. However, this input-passing scheme also resulted in a growing input size for the model. Since standard Transformers have a quadratic memory complexity, this lead to memory constraints for those intervals that occur later in the games [KWH22]. To overcome the resulting limited context size, we introduced the above-mentioned filtering of the chat messages.

However, training our model was still rather expensive with respect to the GPU VRAM consumed, even if the LLM is kept frozen and only the single projection layer is trained. This is attributed to the need to store the intermediaries of the LLM forward pass for the gradient computation of the projection layer in the backward pass. Due to this, setups 1 and 3 have almost the same memory consumption even though the LLM is kept frozen in setup 1. From this perspective, the model architecture is not optimal for memory-efficient training but was chosen for its flexibility and multimodal capabilities. To mitigate this effect and still allow for training on a single NVIDIA V100 GPU with

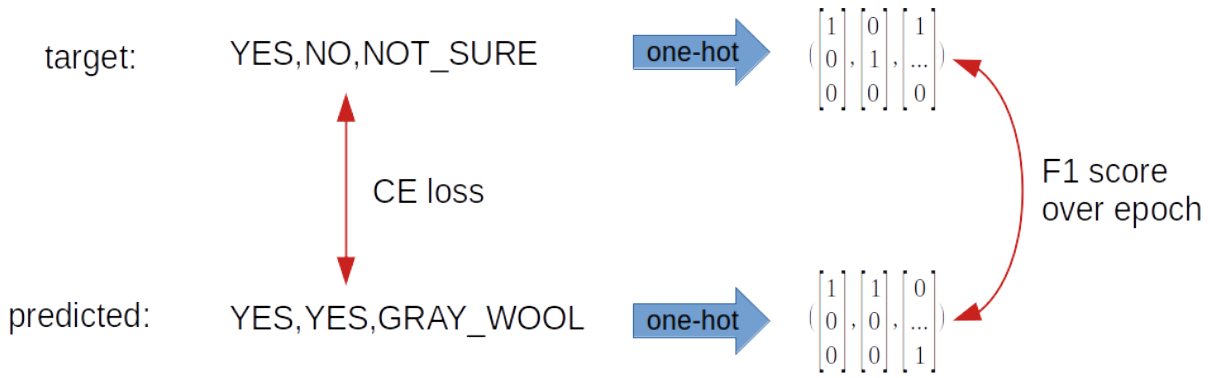

Figure 3.6: Cross Entropy (CE) loss and F1 score calculation for ToM tasks. CE loss is performed on token level. F1 score is calculated for each question on the one-hot encoded output. The size of the vectors is determined by the number of possible answer options.

32GB VRAM, we set the batch size to 1 and used gradient checkpointing [CXZG16]. For fine-tuning the LLM, we applied QLoRA with various $alpha$ and r values (see Chapter 4.1). Using QLoRA additionally reduced the memory consumption during training, as only the quantized weight parameters of the LLM needed to be stored in memory.

Loss and Evaluation Metric We focus here on the evaluation of the model on the ToM tasks, while specifics regarding the CPA tasks are discussed in Section 4.5. For the ToM tasks we expect the model to answer the probing questions with a single word each, separated by a comma. The model's output is then compared to the ground truth answers and the loss is calculated on token level. We use the Cross-Entropy (CE) loss described in equation 3.1 for backpropagation during training, where w_t represents the token at position t and $P(w_t|w_{1:t-1})$ is the probability of predicting w_t given the preceding context [LHH+24].

$$\text{CE Loss} = \frac{1}{T}\sum_{t=1}^{T} -\log(P(w_t|w_{1:t-1})) \tag{3.1}$$

For evaluation and comparison against the baseline, we calculate the F1 score for each epoch. For this, the natural language output of the model is separated into the three answers and one-hot encoded with respect to the possible answer options per question. As illustrated in Figure 3.6, the F1 score is calculated for each question per epoch using those one-hot encoded vectors. In line with the baseline paper, we compute the F1 score as a multi-class average F1 score weighted by the number of instances in each class [BCC21].

When categorizing the model's output into the possible answer options, the model's output string must exactly match the ground truth answer. Using string similarity metrics like Levenshtein distance would be possible to come closer to how a human would understand such answers. However, this was not done in this work, as the model mostly spelled the words correctly and exact matching already showed sufficient performance.

Another point to mention is that the CE loss is calculated based on the token level output of the model's forward function, while the F1 score is calculated based on the output of the generate function. This can lead to a mismatch between the loss and the F1 score. The model could generate the correct answer for the F1 score evaluation, but the loss can still be high if the token level output is not correct. Furthermore, we call the generate function only every n iterations to save training time, while the forward function is called in every iteration. This leads to a sampling bias in the F1 score calculation in comparison to the loss curve. However, for the test set the generate function is called in every iteration to get the precise F1 score.

4 Experiments

4.1 Hyperparameter Tuning

With the above-described architecture and training regime in place, we conducted several experiments to evaluate the performance of our model on the *Mindcraft* dataset. For that, finding suitable hyperparameters was the first step. Those tests were conducted with model setup 2 (text only) on the ToM tasks and on a subset of the dataset with half of the size. Throughout the experiments, we fixed the number of training epochs at 10. The following hyperparameters were tuned independently of each other:

LoRA alpha and r First, we used the commonly suggested ratio of $alpha = 2 * r$, which is also used in the original LoRA paper [HSW+21], and varied the magnitude of the values. We adjusted the values between $r = 4$ and $r = 256$ in four steps, with $alpha = 8$ and $alpha = 512$ respectively.

In the next step, we held the magnitude of $alpha$ at 8 and altered r between 4 and 32 in four steps, testing the optimal ratio for this task.

LoRA dropout rate LoRA specifies a dropout rate for the attention weights, which is set to 0.1 in the original paper. We tested the performance of the model with dropout rates of 0.05, 0.1, 0.3, and 0.6.

Learning rate The learning rate is varied between 7×10^{-4} and 7×10^{-6} in four logarithmic steps. As this is evaluated with model setup 2, the learning rate only applies to the LoRA layers and does not influence the projection layer, which is not in use here.

Weight decay Weight decay describes the technique of adding a regularization term to the loss function for penalizing large weights. This helps in preventing overfitting and promoting a simpler, more generalizable model. In our experiments we changed the weight decay between 1×10^{-3} and 1×10^{-7} in five logarithmic steps.

Projection dropout rate This variation was conducted with model setup 3 to include the video modality and the projection layer. We added a dropout layer directly after the projection layer, for which we tested the dropout rates of 0.0, 0.2, 0.4, and 0.6.

4.2 Importance of Alignment Learning

In the work of Wu et al., a separate stage of alignment learning is used to align the different modalities with the text feature space. This approach is in line with the approach of other works on MM-LLMs to isolate the feature space alignment from the task-specific fine-tuning process [LLWL23; WFQ+23; ZLB23]. However, we wanted to test the importance of separate alignment learning for our model and task combination. Therefore, we conducted experiments with and without explicit alignment learning to compare the performance. Speaking in the language of the above-introduced model setups (see Table 3.1), this comparison is between setup 3 training basing upon setup 1 and setup 3 training without prior training.

4.3 ToM Tasks

As the main experiment of this work, we evaluated the performance of our model on the ToM tasks depicted in Section 3.1. For that, we explored the behavior of the different model setups from Table 3.1 before finally comparing the best setup of our model with the baselines of Bara et al. [BMY+23] and Bortoletto et al. [BRA+24].

We perform this experiment on the full dataset, with the train/test split from the original paper and the best hyperparameters found in the previous step. In contrast to the original Mindcraft model, our model is not a time-series model with information persistence. Therefore, the experimental setup varies slightly. Their LSTM model processes chat messages or video frames every second and stores the information in the cell states. In timesteps where the questions are asked, the model is probed with the questions in addition to the new input, attending to the stored states from the past to predict the answers [BCC21]. However, LLMs are not specifically designed for time-series data and therefore our model processes the full interval input together with the questions as explained in Section 3.2. With this setup, the baseline model is "following" the game in a more intuitive way instead of reacting upon all the information at once.

Further difference lies in the way the information about the dialogue moves is used. Bara et al. explicitly designed the prediction of the dialogue moves as a separate pre-training stage and then made use of that explicit prediction as an input in the ToM tasks [BMY+23]. In contrast, our model never sees or predicts the dialogue moves explicitly, but the dialogue moves are only used to filter the chat in pre-processing.

Those differences in the experimental setup are to be attributed to the different nature of the underlying models but are believed to still preserve the core of the evaluation tasks.

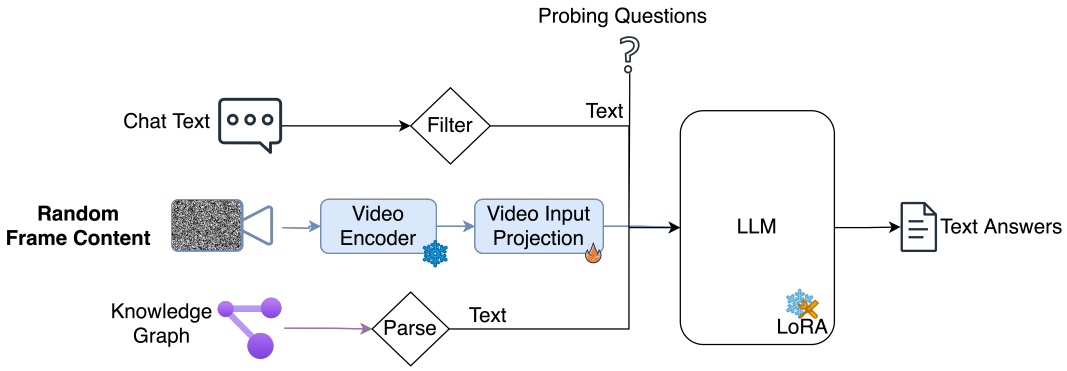

Figure 4.1: Architecture for testing the importance of the video modality. The real video input is replaced with random noise.

4.4 Importance of Video Modality

One of the main research questions of this work is "How important is multimodality for ToM tasks?". To answer this question, we performed two analyses within this work. As a first step, we compared the performances of model setup 2 (text only) and setup 3 (text and video) as part of the preceding experiment. Additionally, we wanted to directly measure the contribution of the content of the video modality to the model performance.

To have a training setup with minimal differences, we designed an experiment where the video modality was replaced with random noise. Instead of encoding the actual sampled video frames with the video encoder, random arrays with the same size and value range were instead injected (see Figure 4.1). With this input modeling, we executed a batch of tests with model setup 3, and compared the results with the results of the real video input.

4.5 CPA Tasks

For having a common understanding of the possible collaborative actions, it is crucial to have an understanding of the missing knowledge of yourself and the other agent. Therefore, we also explored the capabilities of our model architecture towards this task of Collaborative Plan Acquisition (CPA).

As described in Section 3.1, the problem is divided into the two tasks of inferring its own and the partner's missing knowledge. In contrast to the combined prompt for the ToM tasks, we tested these tasks separately within the experimental setup. This decision

originated from the more complex answer structure of the CPA tasks and the fewer training samples to be trained on. Fewer training samples are present because the CPA tasks are evaluated at the end of the game, which reduces the testable intervals to one per game. With the two CPA tasks tested separately, more granular and less error-prone evaluation was achieved.

The prompt structure used for the CPA tasks is very similar to the prompt for ToM tasks (see Listing 4.1). The system prompt, filtered dialogue chat and own material knowledge are the same. Only the question paragraph is different, asking for the missing knowledge of the agent instead of the mental state of the partner. In Listing 4.1, the prompt for the CPA task is shown in which the model is asked for the missing knowledge of the partner. Respectively for predicting its own missing knowledge, the model would be probed with "Which recipes for building blocks does your partner have in their plan, but you don't?" instead.

```
You collaborate with a human in a 3D game like Minecraft, aiming to create a goal
material. Both of you lack knowledge and must exchange information to reach your goal.

Dialogue:
Human: i have an golden axe
You: i have diamond and; gray wool + golden axe
Human: emerald bloc = diamond block + blue wool

Own Material Knowledge:
IRON BLOCK + LAPIS BLOCK with GOLDEN AXE = SOUL SAND
CYAN WOOL + GRAY WOOL with GOLDEN AXE = IRON BLOCK
...
OAK PLANKS with IRON AXE = DIAMOND BLOCK

Question:
Which recipes for building blocks do you have in your plan, but your partner doesn't?
Use the answer format as used above in the plan. Do not explain your answer.
If more than one recipe, separate by a comma.
```

Listing 4.1: Prompt example for CPA tasks. The video embedding is left out for clarity.

Similar to the ToM tasks, the experimental setup differs from the baseline of Bara et al. due to the nature of the model architecture. Again, we do not pass the game information to the model in a time-series manner but all at once when the CPA task is being probed. Moreover, they used the same model component which explicitly models the ToM tasks as a basis for the CPA task (see Figure A.2) [BMY+23]. This stands in contrast to our approach, where the models performing on ToM and CPA tasks share the same architecture (see Figure 3.4) but are different instances with different model parameters that do not depend on each other. As a result, we do not explicitly model the intermediate steps towards Collaborative Plan Acquisition and instead rely on the

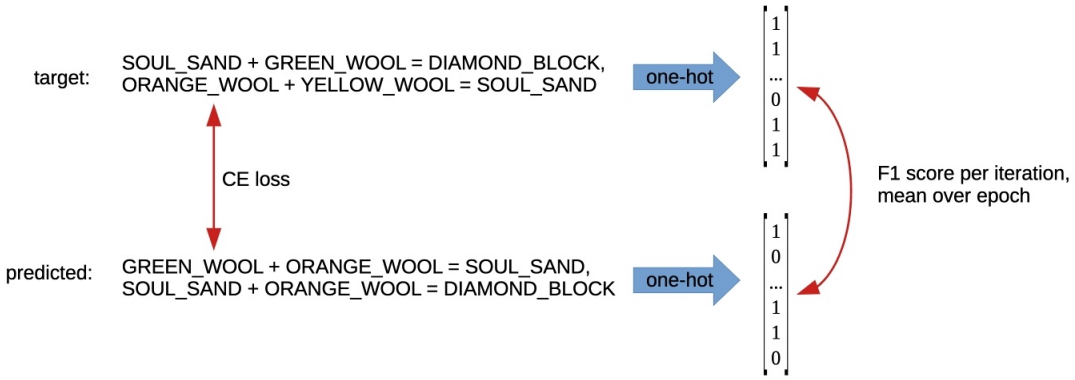

Figure 4.2: Cross Entropy (CE) loss and F1 score calculation for CPA tasks. CE loss is performed on token level. F1 score is calculated on the edges of the knowledge graph, represented as one-hot encoded vectors of all possible edges.

model learning those implicitly. However, Bara et al. also only probe the CPA problem to the model once at the end of each game. This ensures that the same number of samples are available between the two approaches.

While the focus within Section 3.4 lies on the evaluation of the ToM tasks, the differences and specifics in evaluation concerning the CPA tasks shall be briefly outlined. Based on the prompt shown in Listing 4.1, the model is expected to answer with the missing recipes for building blocks in the plan of the partner or its own. This answer should follow the format depicted in Figure 4.2 and should not include any explanations. The loss function used for training the model on the CPA tasks is the same as for the ToM tasks, Cross-Entropy loss on token level. However, the way how we calculate the F1 score differs slightly.

The output is parsed and split into the edges that are resembled by it. For example "SOUL_SAND + GREEN_WOOL = DIAMOND_BLOCK" would be split into the two edges (SOUL_SAND, DIAMOND_BLOCK) and (GREEN_WOOL, DIAMOND_BLOCK). As the knowledge graph is a directed graph, the order of the materials does matter here. The edges from the prediction and target are then converted into a one-hot encoded vector respectively, where the size of the vector is defined by all possible combinations of materials in the game (see Figure 4.2). Those vectors are used for the F1 score calculation per each game. We calculate the mean across all games in the epoch and report this as the final F1 score for the epoch.

This evaluation method for the predicted edges is in line with the evaluation in the baseline of Bara et al., even though the way of retrieving the edges from the model differs due to the nature of the model. By extracting the edges from the model output

in this way, the score is not affected by which recipe or material is stated first in the answer but just by the information content. Similar to the ToM tasks, we conducted the evaluation on the full dataset, with the train/test split from the original paper and with the best hyperparameters found in the hyperparameter tuning experiment.

4.6 One-Shot Prompting

Prompting techniques such as few-shot learning were shown to improve the ToM abilities of LLMs significantly [ZNB21]. Therefore, we compared the performance of our model setup 0 with a zero-shot prompt (see Section 4.3 and Section 4.5) with the performance of the same model when prompted with a one-shot prompt. This experiment was conducted for both the ToM and CPA tasks.

Ultimately, this experiment was constructed with the research question in mind, whether the model can reach similar performance levels with one-shot prompting as with fine-tuning. Therefore, we designed the one-shot prompt to provide the model with the intended structure of the answer so that it can directly adhere to it with the help of in-context learning.

The example used in the one-shot prompt was selected manually to be of high quality and to provide the model with the necessary information to infer the reasoning behind the shown example. To ensure that the example comes from the same distribution and shows realistic behavior, we selected the instance from the training set of the Mindcraft dataset. This of course gives a slight advantage for the training score, but should only have the intended zero-shot effect in the test evaluation, as the game from the example is not part of the test set. We decided to show a full example including chat and knowledge graph within the prompt instead of only appending a demonstrative answer to the original prompt. By this means, it follows more the structure of a chat-like conversation on which the model has been instruction-tuned on. Moreover, we assumed that the model can better understand the context of the answer and the reasoning behind it when the full context is shown.

The example is inserted between the system prompt and the real dialogue chat for the ToM and CPA tasks respectively (see Listings 3.1 and 4.1) and is prepended with the note that it is an example (see Listing 4.2). We selected the example dialogue chat, knowledge graph and ground truth answers to be from the last interval in a game, which makes it usable for both ToM and CPA tasks.

Listings 4.2, 4.4 and 4.3 show the example prompts for the ToM and CPA tasks respectively. To emphasize the multi-turn conversation structure, the model-specific prompt template is also included.

```
<s>[INST] You collaborate with a human in a 3D game like Minecraft, aiming to create a
    goal
material. Both of you lack knowledge and must exchange information to reach your goal.
This is an example:

Dialogue:
You: we need gold block; made from cyan wool + redstone block; here's redstone block;
    break with golden shovel
Human: do you know how to make cyan?
You: orange wool + green wool; don't know how to make orange wool though
Human: blue + red is orange
You: blue is green + cobble
Human: red is emerald + redstone
You: okay so we need cyan; so we need orange + green
Human: we have red now; red + blue = orange; how do we make blue again?
You: green + cobble
Human: blue + red = orange
You: okay now we need red
Human: hoe to mine blue? how does one make cyan?
You: now orange + green = cyan

Own Material Knowledge:
CYAN_WOOL + REDSTONE_BLOCK with GOLDEN_SHOVEL = GOLD_BLOCK
ORANGE_WOOL + GREEN_WOOL with DIAMOND_HOE = CYAN_WOOL
JUNGLE_PLANKS with GOLDEN_SHOVEL = REDSTONE_BLOCK
ACACIA_PLANKS with GOLDEN_SHOVEL = GREEN_WOOL
GREEN_WOOL + COBBLESTONE with DIAMOND_HOE = BLUE_WOOL
BIRCH_PLANKS with GOLDEN_SHOVEL = COBBLESTONE
SPRUCE_PLANKS with GOLDEN_SHOVEL = EMERALD_BLOCK

Questions:
Has the other player created RED_WOOL until now? Options: NO,MAYBE,YES
Do you think the other player knows how to make ORANGE_WOOL? Options: NO,MAYBE,YES
What do you think the other player is making right now? Take options from own material
    knowledge. Additionally NOT_SURE is an option.

Do not explain your answer. Restrict each answer to the three questions to one word
    and separate them by a comma. [/INST] YES,YES,CYAN_WOOL</s>[INST]
This is the real data and questions:
...
```

Listing 4.2: ToM one-shot prompt including prompt template

We selected this one-shot example as it contains all the necessary information to infer the mental state of the partner and ultimately the answer to the questions. The *Task*

Status question can be inferred from "Human: we have red now", the *Task Knowledge* question from "Human: blue + red is orange" and the *Task Intention* question from the last two lines of the chat (see 4.2). Moreover, the CPA tasks can be inferred from the chat as well, as the agents are discussing the respective missing recipes openly in the chat (see Listings 4.2, 4.3 and 4.4).

```
....
Questions:
Which recipes for building blocks does your partner have in their plan, but you don't?
Use the answer format as used above in the plan. Do not explain your answer. If more
    than one recipe, separate by a comma. [/INST]
REDSTONE_BLOCK + EMERALD_BLOCK = RED_WOOL,RED_WOOL + BLUE_WOOL = ORANGE_WOOL</s>[INST]
This is the real data and questions:
...
```

Listing 4.3: CPA one-shot prompt question for predicting own missing knowledge

```
...
Questions:
Which recipes for building blocks do you have in your plan, but your partner doesn't?
Use the answer format as used above in the plan. Do not explain your answer. If more
    than one recipe, separate by a comma. [/INST]
ORANGE_WOOL + GREEN_WOOL = CYAN_WOOL,GREEN_WOOL + COBBLESTONE = BLUE_WOOL</s>[INST]
This is the real data and questions:
...
```

Listing 4.4: CPA one-shot prompt question for predicting partner's missing knowledge

4.7 Cross-Evaluation

The final experiment conducted in this work is the cross-evaluation of models which were fine-tuned on the ToM tasks on the CPA tasks. This experiment is designed to explore, if explicit modeling of the partner's mental state through ToM tasks is beneficial for the CPA tasks or if the implicit capabilities of the model are sufficient.

For this experiment, we used the fine-tuned models from the ToM tasks in setup 3 (see Section 4.3) and prompted them with the CPA questions (see Listing 4.1). We performed no fine-tuning on the CPA tasks. For comparability, we evaluated on the same test set as the above CPA experiments. The parsing and F1 score calculation was done in the same way as described in Section 4.5.

5 Results

5.1 Dataset Statistics

In order to set the following results into context, we first want to explore the circumstances given by the dataset. The Mindcraft dataset contains 162 recorded games, which are split up with a ratio of 60/20/20 into train/validation/test sets. Each game is separated for the ToM tasks into an average of 4.43 intervals and contains on average 70.77 messages per game from the game chat (see Figure A.3 and Figure A.4).

Moreover, it is worth taking a look at the distribution of the ground truth answers for each of the ToM tasks. Here, we will be looking at the test split from the dataset. When taking a look at the *Task Status* question (Q1), one can see that "Yes" and "No" are occurring with similar frequency while "Maybe" is occurring rarely. However, for question 2, the *Task Knowledge* question, "Yes" is by far the most common correct answer (see Figure 5.1). This can lead to already high scores if the model would only answer "Yes" all the time. For the third question the labels are fairly balanced between the materials, but "NOT_SURE" is the most frequent label by a considerable degree (see Figure A.5). As a result, the random baseline and majority baseline for this dataset are

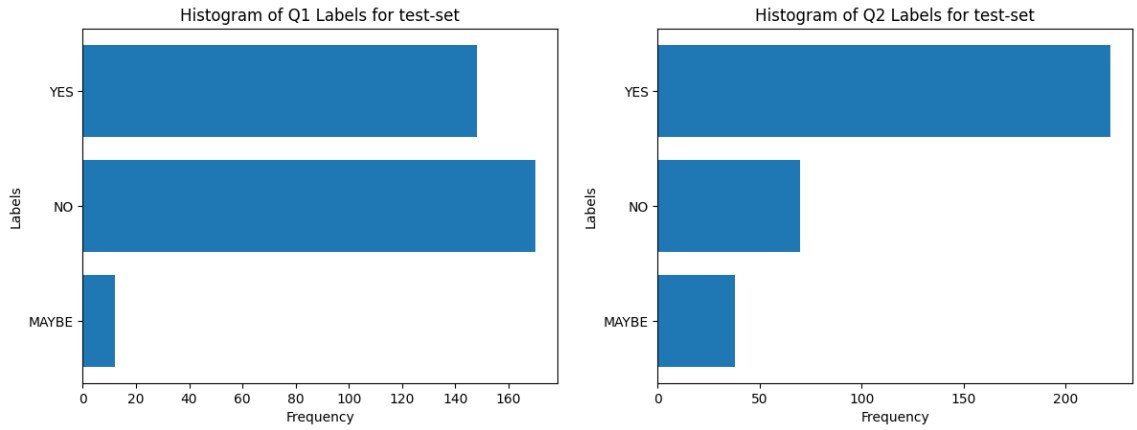

Figure 5.1: Test set label distribution for task status (Q1) and task knowledge (Q2) of the ToM tasks

	Task Status	Task Knowledge	Task Intention
Random Baseline	38.0	37.5	5.0
Majority Baseline	35.0	54.1	5.3

Table 5.1: Random and majority baseline F1 scores

as shown in Table 5.1. We calculated the random baseline as the average achieved score out of 100 random runs. The majority baseline is computed by using the most likely answer option (see Figure 5.1) as the prediction for all instances. As expected, the F1 score of the majority baseline for Q2 is already fairly high by just answering "YES" every time.

5.2 Hyperparameter Tuning

While the technique of hyperparameter tuning is not the focus of our research, finding a hyperparameter set that works well for our model and task combination is still important to compare competitive results with the baselines. Therefore, we briefly report our results here for transparency. It also showcases the robustness of the model to some hyperparameters and the sensitivity to others. We performed these experiments sequentially, using the best hyperparameters from the previous step for the next one.

LoRA alpha and r We varied the hyperparameters $alpha$ and r of the LoRA layers as described in Section 4.1. As a first approach, we used the ratio between $alpha$ and r from the original paper and varied the magnitude of both parameters. Smaller values with $r = 4$ and $r = 8$ show better scores (see Table 5.2), while increasing to values of 256 seems to lead to learning the wrong features. When taking a look at the token predictions, also the worst-performing variation adheres to the required structure. Therefore, a low F1 score actually correlates to low performance in the ToM tasks. As $alpha$ is the scaling factor for the LoRA block, this could imply that with larger $alpha$ the LoRA block learns features that are simpler to learn with its limited parameters and then overrules the original LLM reasoning. Therefore, using such a factor of $alpha = 2 * r$ seems to be undesirable for this task.

As a next step, we varied the ratio between $alpha$ and r with $alpha$ kept stable at $alpha = 8$. In general, there seems to be robustness against the variation of this ratio or the magnitude of r. Since the parameter r correlates to the number of parameters added with the LoRA block, it is assumed that for this task adaption no fundamental features have to be learned in addition to the existing LLM abilities. From the results in Table 5.2, it can be seen that the best-performing ratio across all tasks is $alpha = 8$ and $r = 32$.

lora_alpha	lora_r	Task Status	Task Knowledge	Task Intention
512	256	44.5	54.4	1.3
64	32	57	55	17.4
16	8	67.1	56.9	27.6
8	4	59.9	57.3	32.4
8	4	59.9	57.3	32.4
8	8	65.1	59.5	30.1
8	16	64.1	60.3	34.4
8	32	62.2	60.7	37.2

Table 5.2: F1 scores for variation of LoRA $alpha$ and r on ToM tasks. The best parameter set is highlighted.

lora_dropout_rate	Task Status	Task Knowledge	Task Intention
0.05	63.7	60.5	29.7
0.1	66.9	61.2	39.1
0.3	62.0	60.6	27.6
0.6	66.7	62.9	33.9

Table 5.3: F1 scores for variation of LoRA dropout rate on ToM tasks. The best parameter value is highlighted.

LoRA dropout rate Using the ratio of $alpha = 8$ and $r = 32$ we tuned the dropout rate of the LoRA block. A dropout rate of 0.1 shows the best performance across all tasks (see Table 5.3). This value is in line with the dropout rate used in the original LoRA paper for similar model architectures [HSW+21]. In general, there seems to be a robustness against the variation of this hyperparameter as the scores are fairly similar across all dropout rates.

Learning rate The learning rate was varied in the range of 7×10^{-3} to 7×10^{-6}. Learning rates on the higher end of the spectrum show drastically decreased performance across all tasks (see Table 5.4). This is also underlined when taking a look at the sampled predictions as they mostly do not adhere to the desired structure. These higher learning rates also do not result in loss convergence in training, which leads to the assumption of overshooting the desired minima. The best-performing learning rate for our model and task combination is 7×10^{-5} (see Table 5.4).

Weight decay Variations of the weight decay hyperparameter seem to be less impactful on the performance of the model than other hyperparameters (see Table 5.5). However,

learning_rate	Task Status	Task Knowledge	Task Intention
7×10^{-3}	1.8	0.5	0.0
7×10^{-4}	46.4	53.7	0
7×10^{-5}	64.0	58.2	29.7
7×10^{-6}	66.4	54.7	24.8

Table 5.4: F1 scores for variation of learning rate on ToM tasks. The best parameter value is highlighted.

weight_decay	Task Status	Task Knowledge	Task Intention
1×10^{-3}	65.9	58.0	33.2
1×10^{-4}	66.6	55.5	26.0
1×10^{-5}	60.1	55.4	21.4
1×10^{-6}	66.6	59.1	34.3
1×10^{-7}	61.9	58.7	29.7

Table 5.5: F1 scores for variation of weight decay on ToM tasks. The best parameter value is highlighted.

a weight decay of 1×10^{-6} shows the best performance across all tasks and is therefore taken for further experiments.

Projection dropout rate For the projection layer between the video encoder and the LLM, a dropout rate of 0.4 shows the best performance across all tasks (see Table 5.6).

We briefly want to mention that we performed the hyperparameter tuning on a subset of the whole dataset and therefore the results are not directly comparable to the final scores. Moreover, the experiments are only performed once per variation. Therefore, the results should be taken with caution as only tendencies can be observed with such a setup.

Finally, the best set of parameters for our model and task combination are:

- LoRA: $alpha = 8$, $r = 32$, $dropout\_rate = 0.1$
- Learning rate: $lr = 7 \times 10^{-5}$
- Weight decay: $wd = 1 \times 10^{-6}$
- Projection dropout rate: 0.4

These are used for all further experiments in this work.

projection_dropout	Task Status	Task Knowledge	Task Intention
0.0	62.9	54.7	23.9
0.2	63.4	56.1	24.8
0.4	65.2	63.5	38.4
0.6	60.2	56.1	26.9

Table 5.6: F1 scores for variation of projection dropout rate on ToM tasks. The best parameter value is highlighted.

	Task Status	Task Knowledge	Task Intention
Setup 3	64.8 ± 0.9	58.8 ± 0.6	32.6 ± 2.1
Setup 3.1	66.3 ± 1.6	59.8 ± 0.3	34.9 ± 3.7

Table 5.7: Importance of alignment learning (Setup 3: random initialization of projection layer, Setup 3.1: setup 3 training based on already aligned projection layer). The F1 scores on the ToM tasks are shown.

5.3 Importance of Alignment Learning

Based on these hyperparameters we evaluated the importance of alignment learning for the ToM tasks. We denote the combination of LLM LoRA fine-tuning and projection layer training (Setup 3) based on the already fine-tuned projection layer (Setup 1) as "Setup 3.1" in Table 5.7. This stands in contrast to the LLM LoRA fine-tuning and projection layer training only based on the pre-trained LLM and with random initialization of the projection layer (Setup 3, see Table 3.1).

The results show a slight improvement in the ToM tasks when doing the full fine-tuning based on the already aligned projection layer (see Table 5.7). However, the results are not significantly different from each other, with the standard deviations overlapping for two of the tasks. This implies that the isolated alignment learning is not as crucial in our setup as the fine-tuning of the LLM. Moreover, it seems that the alignment of the video modality can happen mostly in parallel to the LLM fine-tuning. This might be due to the limited number of modalities used here and would explain why modality-specific alignment learning is still frequently used in other multimodal research with more modalities [WFQ+23].

We decided therefore to use the simpler training setup for the following experiments in this work, as it uses approximately 50% less resources with the results not being significantly different from the more complex setup.

5.4 ToM Tasks

With the parameters and insights from above, we evaluated the ToM tasks in the different model setups. We compare the performance of the different model setups on the ToM tasks to explore the influence of fine-tuning in opposition to zero-shot prompting. Finally, we compare the results to the baselines from Bara et al. and Bortoletto et al.

Setup 0 The experiments show that only using the zero-shot prompt depicted in Listing 3.1 without fine-tuning on the text modality performs worse than the random baseline (see Setup 0 in Table 5.8). When analyzing the predictions, we could observe various tendencies. The expected answer formatting explained in the prompt is not observed in roughly half of the cases. Most commonly the answers are given in numbered lists, explained further or even more than three answers are given. These formatting issues impair the simple parsing mechanism and invalidate the answers as wrong predictions. However, even when analyzing the predictions manually, only few of the wrongly formatted predictions would actually be semantically correct. Therefore, the fundamental result would not be altered if the syntactical errors were factored out.

Notably, we observed a strong bias towards answering "NO, MAYBE, ..." for questions 1 and 2 in the predictions of model setup 0. In one experiment for instance, 283 of 329 predictions for the task status question were "NO". In the same experiment 190 of 262 predictions for task knowledge were equivalent to "MAYBE". These two predictions were observed to occur frequently in combination with each other. For the third task about task intention, the model predicted the answer option "NOT_SURE" more often than its occurrence in the data distribution (see Figure 5.1), but this imbalance was not as extreme. Moreover, the third answer was often invalidated by explanations being appended.

These observations lead to the interpretation that the model was so intensively pre-trained on data, where explanations are crucial, that just simply prompting it to not explain is not a sufficient incentive. Moreover, we hypothesize that "NO, MAYBE, NOT_SURE" seems to be the safest and at the same time most likely answer option without the model having seen the data distribution to predict for.

Setup 1 In comparison, Setup 1 shows already significantly better performance. Our model setup 1 already scores better across all tasks than the baseline from Bara et al. and similar to the GNN alteration from Bortoletto et al. (see Tables 5.8 and 5.9). Both of these baselines would be comparable to our model setup 3 regarding the training setup and modalities. In comparison to our setup 0 results, this performance difference could be explained by the added information content of the video modality or by some

	Task Status	Task Knowledge	Task Intention
Setup 0	31.2 ± 1.0	15.3 ± 0.2	0.9 ± 0.7
Setup 1	56.8 ± 7.5	54.4 ± 3.2	21.0 ± 3.3
Setup 2	65.1 ± 1.4	60.0 ± 3.2	31.0 ± 0.9
Setup 3	64.8 ± 0.9	58.8 ± 0.6	32.6 ± 2.1

Table 5.8: F1 scores of our model on the ToM tasks with different model setups

	Bara et al. (2023)	Bortoletto et al. (2024)	Human	Ours
Status				
P+D	45.5 ± 2.3	59.1 ± 0.6	67.0	**65.1 ± 1.4**
P+D+V	45.2 ± 1.8	58.9 ± 0.8	67.0	**64.8 ± 0.9**
Knowledge				
P+D	50.0 ± 1.5	57.2 ± 1.5	58.0	**60.0 ± 3.2**
P+D+V	50.2 ± 1.1	57.5 ± 1.7	58.0	**58.8 ± 0.6**
Intention				
P+D	8.7 ± 2.1	11.1 ± 1.8	46.0	**31.0 ± 0.9**
P+D+V	10.5 ± 2.3	12.1 ± 2.4	46.0	**32.6 ± 2.1**

Table 5.9: F1 scores on ToM tasks with different modalities (P+D: Plan + Dialogue, V: Video). Our results are compared against the baselines [BMY+23; BRA+24] and human performance. We used model setup 2 for modalities P+D and model setup 3 for P+D+V.

implicit "prompting" happening through the training of the projection layer. We evaluate these hypotheses in further experiments in this work.

In the analysis of the predictions for model setup 1, we observed that already after half an epoch the model adhered to the instructed answer format, even though the LLM is not fine-tuned itself. Moreover, the model shows no obvious hallucinations of materials that are not mentioned in the prompt. Therefore, the reported score correlates to real ToM task performance in this setup. These observations would support the former hypothesis of implicit "prompting" happening by the trained projection layer.

Setup 2 and 3 Model setups 2 and 3 show very similar performances within each other's standard deviation (see Table 5.8). Both setups perform hereby better than both baselines across all ToM tasks and even reach human performance levels in two of three

ToM tasks (see Table 5.9). In the most challenging third task about the agent's intention, our model outperformed the baseline scores by far and took a big leap towards human performance levels. This implies that fundamental Theory of Mind abilities are present in our LLM-based model and that fine-tuning helps in making these abilities usable for the ToM tasks. However, the similarity in scores for model setups 2 and 3 raises the question of how important the video modality really is for solving the tasks at hand.

In conclusion, the experiments show the importance of fine-tuning LLMs in comparison to zero-shot prompting for ToM tasks. With fine-tuned MM-LLMs higher ToM scores than current baseline models can be achieved, which shows promising directions for further research in collaborative environments.

5.5 Importance of Video Modality

Based on the research question about the importance of multimodality and the results from the prior experiment we performed this experiment to test the importance of the video modality for this task. As described in Section 4.4, the training setup is the same for both variations except that one group gets random noise instead of the real video frames as input.

As it can be seen in Table 5.10, using random video input instead of real frames showed no significant difference in the performance. We observed especially no performance deterioration but scores within each other's standard deviation. Apparently, the output of the video encoder holds no meaningful content for the ToM tasks. Possibly, the video encoder being used focuses on other features that are not relevant to this task. While the general insignificance of the video modality for this task seems unlikely, the limited quality of the video frames and the sampling method used are assumed to have the highest influence on these results.

Before the frames from the game are passed to the video encoder, they are sampled evenly into a fixed number of clips of two seconds duration each. This method is specific to the ImageBind encoder and originates from their original paper [GEL+23]. Additionally, when looking at the frames from the dataset, some frames do contain information that seems relevant to the human eye for this task, but the majority of frames do not. Combining now this sparsity of meaningful content in the dataset with the very simple sampling method, it seems plausible that the video encoder could not grasp any meaningful information.

For the tasks on which ImageBind and other encoders were trained such simple sampling methods seem to be a sufficient balance between performance and complexity. However, for task-oriented collaboration the importance of frames is not as evenly distributed,

	Task Status	Task Knowledge	Task Intention
Setup 3, real frames	64.8 ± 0.9	58.8 ± 0.6	32.6 ± 2.1
Setup 3, random noise	66.1 ± 2.9	60.3 ± 3.3	33.2 ± 2.9

Table 5.10: Importance of video frame content for ToM tasks. The F1 scores of our model trained with real video frames and random noise are compared.

but only a small amount of frames contain the most meaningful actions. For further advancements in multimodal collaborative tasks more intelligent sampling methods should therefore be used for filtering out the meaningful actions, e.g. as already used in the field of action classification [BPM22]. Furthermore, this instance confirms the common paradigm of taking great care of the data quality, because the model quality strongly depends on it.

5.6 CPA Tasks

Similar to the ToM tasks, the CPA tasks are evaluated first by analyzing the performance of the different model setups before comparing our model with the baseline performance. As discussed in Section 4.5, the CPA tasks are evaluated by comparing the predicted edges with the ground truth edges missing in the knowledge graph. The two CPA tasks, predicting the partner's missing knowledge and predicting their own missing knowledge, are evaluated and reported separately.

Setup 0 Model setup 0 shows very low scores for both CPA tasks (see Table 5.11). A main reason for the low scores are the formatting issues in the predictions. Often explanations are given or different formats such as numbered lists and newlines are used for the answer. Moreover, the tools and mines are sometimes mentioned as part of the recipe. Mines are these materials in the game that can be converted into another material without a second material (see for example "BIRCH_PLANKS" in Listing 4.6). However, these edges are never missing in the partner's or own knowledge graph and should therefore never be predicted. These prediction faults demonstrate the imprecision in the given prompt in which we did not explicitly tell the model to ignore the tools and mines.

Putting these formatting issues aside, the predictions are still often semantically wrong. Additionally, more than one missing recipe is predicted frequently in cases where only one might be missing in the agent's knowledge. It is already visible in these results that

	Partner's Knowledge	Own Knowledge
Setup 0	11.0 ± 1.4	1.6 ± 0.4
Setup 1	47.0 ± 7.7	31.4 ± 16.9
Setup 2	80.0 ± 2.5	74.7 ± 5.6
Setup 3	78.8 ± 3.3	75.6 ± 6.6

Table 5.11: F1 scores of our model on the CPA tasks with different model setups

predicting its own missing knowledge is more difficult than predicting the partner's missing knowledge.

Setup 1 Setup 1 shows a significant improvement in the CPA tasks compared to setup 0. The model follows the answer format in most cases and provides no explanations in the answers. This behavior is consistent with the ToM tasks, which makes the assumption of implicit "prompting" through the projection layer more likely. The tools and mines are still sometimes part of the recipe, but this is not as often the case as in setup 0. Moreover, the predictions are semantically better and the frequency of giving too many recipes decreased. The high standard deviation in the scores for model setup 1 is attributed to the sensitivity of the simplistic result parsing.

Two interesting behaviors are observed in the predictions for setup 1. First, the model hallucinates about recipes that do not exist in the dataset, have never been observed or sometimes would not even make sense. Examples for this are "GOLD_BLOCK = DIAMOND_BLOCK + 2*GREEN_WOOL", "COBBLESTONE + DIAMOND_PICKAXE = ACACIA_PLANKS" or "CYAN_WOOL = ACACIA_PLANKS". The semantic errors in these errors consist of materials being used more than once in a recipe where this was never part of any observed recipe in the dataset, mines being used as target materials, tools being used as materials or recipes only consisting of a single source and target material.

The second interesting behavior is that the model does not only change its prediction format but also apparently discards faulty natural language behavior which it learned during LLM pre-training. The specific LLM used for our model (Mistral 7B) is known to escape underscores in the text with backslashes. This behavior is observed in model setup 0 of ToM and CPA tasks, but not in setup 1 of the respective tasks even though the LLM is kept frozen. Since the LLM is not fine-tuned in setup 1, this change in natural language behavior cannot be attributed to unlearning previously learned behavior within the LLM, but rather to the behavior being overwritten by the implicit instructions given through the projection layer. Whether this implicit prompting is actually related to the video modality seems unlikely considering the results from Section 5.5. However, this

	Bara et al. (2023)	Bortoletto et al. (2024)	Ours
Partner's missing knowledge			
P+D+V	66.8 ± 1.5	56.5 ± 0.3	**78.8 ± 3.3**
Own missing knowledge			
P+D+V	28.4 ± 1.4	58.4 ± 0.8	**75.6 ± 6.6**
Overall			
P+D+V	47.6 ± 1.5	57.5 ± 0.6	**77.2 ± 5.0**

Table 5.12: F1 scores on CPA tasks with all modalities (P+D+V: Plan + Dialogue + Video). Our results are compared against the baselines [BMY+23; BRA+24].

hypothesis would need to be confirmed in a respective experiment for setup 1 with random frame input.

Setup 2 and 3 When fine-tuning the LLM in model setup 2, the performance in the CPA tasks is further improved (see Table 5.11). As already in setup 1, the model obeys the instructed answer format. Here however, no tools or mines are part of the recipes anymore, so the model apparently learned to ignore these. The predictions are semantically not far from perfect, while most of the penalty in the score is given because of too many predictions. The big leap in performance from model setup 1 to 2 shows that fine-tuning the LLM does help in getting the right behavior in the CPA tasks.

The performance for model setup 3 is very similar to setup 2 (see Table 5.11). Again here, there are quite some instances of too many or too few recipes given, which leads to a lower score. This indicates that the hidden task of deciding how many recipes are missing might be harder to solve for our model than the actual CPA task of predicting the missing edges. This hypothesis would need to be confirmed in isolated experiments. Nevertheless, also the setup 3 model is not free from hallucinations and reasoning errors as it predicts unreasonable recipes with self-loops like "LIME_WOOL + IRON_BLOCK = LIME_WOOL".

In line with the above experiments, including the video modality in the training process does not show a significant difference in the performance of the CPA tasks (see setups 2 and 3 Table 5.11). All variations also confirm the finding of Bara et al., that predicting one's own missing knowledge is more difficult than predicting the partner's missing knowledge. However, our model significantly reduces the gap between the two tasks, which implies that the advanced reasoning capabilities of LLMs are beneficial here. In the

overall comparison with the baselines from Bara et al. and Bortoletto et al., our model outperforms the best of the respective baselines in all CPA tasks (see Table 5.12).

Notably, our model reaches these scores without explicitly modeling the ToM features. This finding is in line with other research in this field, which shows that LLMs are capable of learning these features on their own [BRSB24; ZZW24]. In the baseline studies, variations across which features should be explicitly modeled were performed and their scores reported. Our scores are evaluated against the best of these variations respectively in Table 5.12.

5.7 One-Shot Prompting

The main goal of the one-shot prompting experiment is to explore, how close we can get to the performance of model setup 2 when using only one-shot prompting without fine-tuning. This would then build a compromise between the high resource consumption of fine-tuning and the low performance of zero-shot prompting shown in the previous experiments.

However, the results show that the performance of the ToM tasks is not lifted to levels of model setup 2 by using one-shot prompting alone (see first section in Table 5.13). When compared to the zero-shot performance, one-shot prompting significantly improves the performance in the ToM tasks 2 and 3, but not in task 1. We observed more adherence to the format of the answers and more semantically correct answers in the predictions with one-shot prompting. The scores of the task intention predictions are still relatively low due to explanations given or unexpected lowercase spellings, but also the material name is semantically often not correct.

To explore the semantic correctness of the predictions and therefore the true ToM abilities as it would be perceived by a human, a semi-manual cleansing of the parsing issues was performed for those runs. These results are reported in the second section of Table 5.13. The results show that the performance is significantly improved across tasks and prompting methods when the parsing issues are factored out. One can also see that the performance improvement attributed to the one-shot prompting is higher for the task status and task knowledge questions than for the task intention question. As formatting issues are now mostly factored out, this leads to the assumption that the given one-shot example might give some hint which is more helpful for the first two tasks than in the more difficult third task.

Table 5.13 also shows that one-shot prompting combined with the cleansing does lift the performance in the ToM tasks significantly closer to the setup 2 performance, especially for the task status and task knowledge questions. It seems that the reasoning required

	Task Status	Task Knowledge	Task Intention
Setup 0, zero shot	31.2 ± 1.0	15.3 ± 0.2	0.9 ± 0.7
Setup 0, one-shot	33.8 ± 1.5	30.4 ± 4.0	6.2 ± 0.7
Setup 0, zero shot, cleansed	35.6 ± 0.9	31.9 ± 0.9	16.2 ± 1.1
Setup 0, one-shot, cleansed	61.9 ± 1.2	50.2 ± 2.6	16.4 ± 1.5
Setup 2	65.1 ± 1.4	60.0 ± 3.2	31.0 ± 0.9

Table 5.13: One-shot prompting results for ToM tasks. We report the F1 scores for zero-shot and one-shot prompting with setup 0 (text only). We compare against the F1 scores when evaluated on predictions that were manually cleansed from formatting imprecisions. Setup 2 (fine-tuned, text-only) is given as a reference.

for those easier tasks is more easily accessible in the LLM than the reasoning required for the task intention question, for which fine-tuning seems to be required. Notably, the cleansed one-shot results already outperform the baseline models in two of the three ToM tasks without needing any explicit training on the tasks (see Table 5.9 and Table 5.13). This shows the inherent reasoning and possibly ToM abilities of LLMs learned during pre-training, when combined with in-context learning from one-shot examples.

We performed the one-shot prompting experiment also for the CPA tasks where we used the same example data but adjusted the questions within the example (see Section 4.6). The results show that the performance in the CPA tasks is not lifted to levels of model setup 2 by using one-shot prompting alone (see Table 5.14). In comparison to zero-shot prompting, the performance for predicting its own missing knowledge is significantly improved, while the performance for predicting the partner's missing knowledge did not show a significant change. As the task of predicting its own missing knowledge is considered to be more difficult, the one-shot example might showcase the reasoning required for this task more clearly than the task formulation itself.

However, the performance of the one-shot prompting is still significantly lower than the performance of the setup 2 model. We observed in the predictions that, similar to the zero-shot predictions, the tool and mines are mentioned often as part of the recipe. Apparently, one example is not sufficient to teach the model to ignore this imprecision in the prompt. Most significantly though, the model fails to recognize knowledge gaps in the partner's knowledge and often predicts "None" as if no knowledge is missing. This behavior is not as prevalent in the predictions for its own missing knowledge but is still present. In comparison to the ToM tasks, formatting issues are not the main issue in the

	Partner's Knowledge	Own Knowledge
Setup 0, zero shot	11.0 ± 1.4	1.6 ± 0.4
Setup 0, one-shot	9.1 ± 1.2	12.5 ± 2.0
Setup 2	80.0 ± 2.5	74.7 ± 5.6

Table 5.14: One-shot prompting results for CPA tasks. We report the F1 scores for zero-shot and one-shot prompting with setup 0 (text only). Setup 2 (fine-tuned, text-only) is given as a reference.

one-shot CPA tasks, but rather the semantic content of the predictions. Therefore, we performed no manual cleansing of the predictions for the CPA tasks. These results imply that the abilities required for recognizing missing knowledge are not easily accessible in the LLM without fine-tuning and cannot be brought out by one-shot prompting alone.

5.8 Cross-Evaluation

The cross-evaluation experiment was performed to explore if models that were previously fine-tuned on ToM tasks have an advantage over using zero-shot prompting for the CPA tasks. This relates to the question if explicit modeling of ToM features is beneficial for related tasks. Therefore, we compare the performance of the fine-tuned ToM model to the performance of the base LLM in the CPA tasks with zero-shot prompting.

The results show that the performance of the CPA tasks is not significantly improved or decreased by using the fine-tuned ToM model for the CPA tasks (see Table 5.15). This is observed across both CPA tasks. Additionally, the predictions show similar faults across both setups.

Apparently, explicit pre-training on ToM tasks does not help in solving related tasks in a zero-shot manner. These results lead to the assumption that the reasoning capabilities required for the CPA tasks are already present in the LLMs and do not need to be explicitly modeled. This is in line with the findings from Zhu, Zhang, and Wang, who showed that general-purpose LLMs already have internal representations for belief states [ZZW24]. However, more experiments and a deeper analysis of the model activations would be needed to confirm this hypothesis. These experiments should make an effort to factor out the formatting and prompting faults to focus the results on the actual ToM capabilities of the models. This has to be left open for future work.

	Partner's Knowledge	Own Knowledge
ToM fine-tuned	8.6 ± 0.9	1.6 ± 0.9
Base LLM	11.0 ± 1.4	1.6 ± 0.4

Table 5.15: F1 Scores of CPA cross-evaluation. We compare the zero-shot performance of the model fine-tuned on the ToM tasks with the base LLM when evaluated on the CPA tasks.

6 Discussion

We now want to discuss the implications of the presented results for the research questions before discussing the limitations of the conducted experiments.

RQ1: Are LLMs capable of performing Theory of Mind tasks in situated dialogues?

RQ2: Do Multimodal LLMs outperform specialized models targeting ToM tasks?

RQ3: How important is multimodality for ToM tasks?

RQ4: Can Multimodal LLMs effectively predict missing knowledge in collaborative situations?

6.1 Theory of Mind Abilities of MM-LLMs

Given the results of the Theory of Mind tasks, we can conclude that LLMs and MM-LLMs are capable of performing Theory of Mind tasks in situated dialogues confidently. The results show that our MM-LLM outperforms the baseline models in all three tasks, with the largest improvement in the most complex task about the agent's intention (see Table 5.9). This indicates that LLMs possess more mature Theory of Mind abilities, especially in the more complex tasks.

Our model is able to outperform the baselines without explicit modeling of Theory of Mind, which suggests already existing agent belief modeling within the pre-trained LLM. In contrast to the baselines, we do not give our model any explicit information about the dialogue moves and pass only the summary of the dialogue as input. With this reduced input, the model is able to achieve close to human performance. This leads to our intuition that the evaluated model possesses foundational Theory of Mind abilities.

In order to reach such performance, the model has to be fine-tuned on the specific tasks. However, we do not attribute this to the model learning the actual Theory of Mind abilities during fine-tuning, but rather to the model understanding the specific task better than our prompt could formulate it. This is supported by the already significantly raised performance when only the projection layer is fine-tuned (see Table 5.8). We furthermore interpret the cleansed results from the ToM one-shot experiment (see

Table 5.13) that the fundamental abilities to model agent beliefs are already present in the pre-trained LLM. Therefore, we see fine-tuning here as just one method next to few-shot prompting to adjust the output format of the model and additionally give the model a better understanding of the task.

While in comparison to the baseline models, our MM-LLM shows a significant improvement in the Theory of Mind tasks, the additional parameter count and computational cost of the model should be considered. Our model has approximately 84 million trainable parameters when using QLoRA, while the LSTM model used in the baseline study has around 18 million parameters. Considering this difference and the need for the 7 billion parameters of the base LLM to be loaded into memory, the computational cost for training our model is significantly higher. For our experiments we fine-tuned the model for 10 epochs, which took around 7 hours on a single GPU. However, already fine-tuning for one epoch showed sufficient performance for the tasks (see Table A.1). Therefore, future work could explore the trade-off between model size and performance in more detail.

Coming back to our third research question, the importance of multimodality for Theory of Mind tasks could not be confirmed in our experiments. The video sampling used for the input to the video encoder did not prove to be suitable for the ToM tasks. In combination with a low density of valuable collaborative information in the video frames, the video modality did not help in improving the performance of the model (see Table 5.10). This observation stands in line with the baseline study of Bara, CH-Wang, and Chai [2021], where the inclusion of the video modality did not show a significant improvement in the ToM tasks for their LSTM model.

When using alignment learning of the video modality as a step before fine-tuning the whole model on the ToM tasks, the performance of the model could be slightly improved (see Table 5.7). However, for our experiments and limited multimodality, the performance difference was not significant enough to justify the additional computational cost of the alignment learning step. We assume that with further modalities added the alignment learning step becomes more relevant.

Additional research could focus on the effect that we observed, where the training of only the projection layer already showed significant performance improvements in the Theory of Mind tasks. Our hypothesis for this is, that the model can implicitly understand the task questions better through the learned embedding than through the limited natural language prompt. Through this effect, our model successfully discarded unwanted natural language behavior that it learned during pre-training.

In conclusion, our results show that LLMs are capable of performing Theory of Mind tasks in situated dialogues and that MM-LLMs outperform specialized smaller models targeting these tasks. We assume that LLMs have learned foundational Theory of Mind

abilities within their pre-training and that fine-tuning on the specific tasks makes those abilities more targeted. However, the importance of multimodality for ToM tasks would need to be further explored with more modalities and better-suited video processing.

6.2 Collaborative Plan Acquisition

To develop effective collaborative abilities, artificial agents need to be able to predict missing knowledge in joint tasks [BMY+23]. This holds for being aware of its own missing knowledge as well as the partner's missing knowledge. For this reason, our fourth research question focuses on the ability of MM-LLMs to predict missing knowledge in collaborative situations.

Our results show that our MM-LLM is able to effectively predict missing knowledge in collaborative settings. More precisely, it outperforms the baseline models in the Collaborative Plan Acquisition tasks (see Table 5.12). While we observed an advantage in performance for predicting the missing knowledge of the partner as well, our model showed the most significant improvement in the more difficult task of predicting its own missing knowledge. Our MM-LLM does not require explicit modeling of the dialogue moves or ToM features in staged trainings as the baseline models but seemingly can make use of its abilities learned during general pre-training. In line with this, the cross-evaluation results showed that no improvements in the CPA tasks can be achieved through explicit pre-training on the ToM tasks (see Table 5.15).

Similar to the Theory of Mind tasks, we fine-tuned the model on the specific tasks to reach such performance. The fine-tuning is hypothesized to mainly help the model in understanding the specific task better, in opposition to actually learning the abilities needed for the task.

Within our results, we could observe that predicting the extent of missing knowledge might be more difficult than predicting the missing knowledge itself. This hidden task was not explicitly modeled in previous works but might be a promising path towards advancing the collaborative abilities of artificial agents.

Concerning our third research question we could not observe a performance increase in the CPA tasks through the additional use of the video modality. Similar to the Theory of Mind tasks, we assume this to be related to the low density of valuable collaborative information in the video embedding being available to the model. Future research with more advanced video processing could show a different outcome.

We conclude that our MM-LLM model is able to effectively predict missing knowledge in collaborative situations. It outperforms the baseline models across both CPA tasks without explicit Theory of Mind modeling.

6.3 Collaboration with MM-LLMs

Our ultimate research goal is to enhance collaboration through the use of multiple modalities and generalized models. This would enable more seamless human-machine interactions if also the model's abilities are sufficient for the tasks. We intuitively assume that multimodality is important for real-world collaboration tasks as it aligns more with human perception and interaction. Moreover, using LLMs for collaboration tasks could be beneficial as they are not limited to fixed question-answer tasks but can interact more intuitively with humans through natural language.

Natural language might be a more intuitive interface for humans, but our experiments showed that it makes it harder to evaluate the model programmatically. This discrepancy is visible in the one-shot results of the ToM tasks, where we performed a semi-manual parsing of the predictions (see Table 5.13). We could show that the LLM can outperform the baselines and reach close to human performance without fine-tuning if the format of the predictions is manually corrected.

The results of our experiments demonstrate that MM-LLMs are showing promising results for collaboration tasks with regard to belief modeling and knowledge prediction. However, real collaboration is more than just the sum of the parts and has to be tested in real-world scenarios with humans to be fully evaluated.

6.4 Limitations

Our work has several limitations that should be considered when interpreting the results. First, the selected prompt can have a significant influence on the model's performance [SCTS23; Ull23]. We did not perform an extensive evaluation of different prompt formats and compare the results of the tasks. Different formats could have influenced the model's performance in the tasks. Moreover, our prompt used for the CPA tasks proved to be more difficult for the model to understand than expected. By this means, imprecise formulations of the prompt were revealed which could have influenced the model's predictions. This left the model with the option to include the tool and mine blocks in the recipe, which was not intended.

Within our hyperparameter search we did not perform a dependent grid or random search as commonly used [BB12], but picked the best hyperparameters independent of each other. A more resource-intensive dependent search could yield better results (see example in Appendix A.5).

While the introduced filtering of the chat messages according to the dialogue moves proved to be beneficial for the memory requirements of the model, it also limited the information available to the model. More advanced solutions such as task-specific summarizations could provide a better balance between memory requirements and information availability.

We also want to point out limitations in our evaluation mechanism. Because the LLM being used (Mistral 7B [JSM+23]) commonly escapes underscores in the predictions, our evaluation mechanism accounts for this by correcting this flaw. However, the model quickly unlearns this behavior during fine-tuning. Moreover, the parsing mechanism used for the evaluation of the predictions is very simple and does not account for alternative wordings or formatting. This had a notable effect on our results as shown in the one-shot evaluation of the ToM tasks (see Table 5.13). We believe however, that the core findings of our work still hold. In future works, a more sophisticated evaluation mechanism should be used to isolate the actual content of the predictions.

Furthermore, our natural language formulation of the recipes in the CPA tasks puts more weight on the target material than on the components of the recipe. Within the baselines, edges were predicted where each directed edge could have a target independent of the other edges being predicted. With our recipe formulation, the target material is the same for both edges, which invalidates both edges if the target material is not correct.

Finally, our work only explored the abilities of one LLM (Mistral 7B [JSM+23]). Further work could therefore include an ablation study with other LLMs to explore the influence of their training data on the development of Theory of Mind abilities. For that, we developed our training framework to be easily extendable to other LLMs, so that future research can easily switch the base LLM used. More advanced and task-specific video processing could also be beneficial for the evaluation of the importance of multimodality.

7 Conclusion

Our work is rooted in the purpose of enhancing human-machine collaboration through the use of multimodal and generalized models. We therefore performed several experiments to assess the Theory of Mind (ToM) capabilities of Multimodal Large Language Models (MM-LLMs) in collaborative settings. Based upon a pre-trained open-source LLM, we developed a multimodal model architecture targeted towards a collaborative environment. The Theory of Mind capabilities of our MM-LLM were evaluated in the context of task-oriented collaboration and situated dialogues within the 3D game environment of *Mindcraft* [BCC21; BMY+23]. There, task status, task knowledge, and task intention of a collaborating agent were predicted by the MM-LLM. We also evaluated the model in Collaborative Plan Acquisition (CPA) tasks, where its own or the other agent's missing knowledge was to be predicted. We see these tasks as a necessary step towards common ground and successful belief reasoning in collaborative settings.

From our experiments, we found that MM-LLMs are capable of predicting the task status and task knowledge of the other agent on human levels. While our model did not reach human performance in predicting the other agent's task intention, it still outperformed previous baseline models. Similarly, our model was able to outperform the more specialized baseline models in predicting its own and the partner's missing knowledge.

Within our experimental setup, we could not confirm our hypothesized importance of multimodality for Theory of Mind tasks. We assume this to be related to the task-unspecific video encoding and the training data used in our experiments. Further research with more task-oriented video encoding is suspected to improve the collaborative performance of MM-LLMs.

In line with related research, our work provides more indications for foundational Machine Theory of Mind capabilities and implicit belief modeling within LLMs. Therefore, we see the use of MM-LLMs as a promising path towards more intuitive human-machine interactions in collaborative settings.

A Appendix

A.1 Chat Filter by Dialogue Moves

To minimize the memory requirements for our model training, we reduced the input token size by filtering the dialogue chat based on the dialogue moves associated with the messages. Each message in the Mindcraft dataset [BCC21] is associated with a so-called dialogue move, which is a categorization of the message's content (see Figure A.1). We selected a subset of dialogue moves that we observed to contain relevant information for the ToM and CPA tasks (see Listing A.1). Based on this subset, we filtered the dialogue chat to only include messages that are associated with one of the relevant dialogue moves (see Listing A.2).

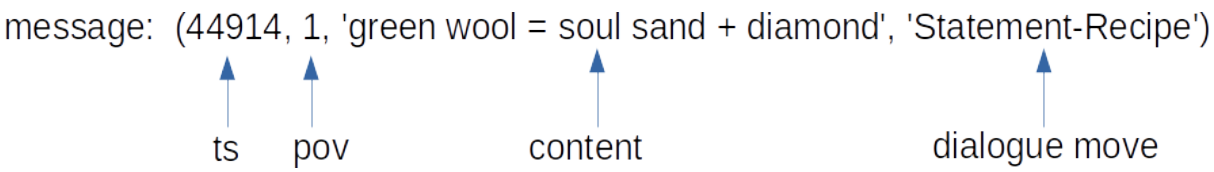

Figure A.1: Metadata information associated with messages in the Mindcraft dataset [BCC21]

['AGREEMENT', 'Directive-Make', 'Directive-Other', 'Directive-PickUp', 'Directive-PutDown', 'Directive-PutOn', 'Inquiry-Act', 'Inquiry-Goal', 'Inquiry-NextStep', 'Inquiry-OwnAct', 'Inquiry-Possession', 'Inquiry-Recipe', 'Inquiry-Requirement', 'Statement-Goal', 'Statement-Inability', 'Statement-LackKnowledge', 'Statement-NextStep', 'Statement-Other', 'Statement-OwnAct', 'Statement-Possession', 'Statement-Recipe', 'Statement-Requirement', 'Statement-StepDone']

Listing A.1: Subset of dialogue move labels considered relevant for the ToM and CPA tasks

You: ok so now its green wool; green wool = soul sand + diamond; do you know how to get diamonds in your tree?

Human: um gray wool has an arrow to diamonds; and soul sand; so i think gray wool + soul sand = diamond

You: ok

Human: so i think gray wool + soul sand = diamond

You: so break the cobblestone with your tool; ok

~~Human: why is it not breaking?~~

You: to make soul sand, you need to break cobble with your tool, place the cobble on the ground; ~~you have to hold left click until it breaks; place it; via right click;~~ break the gray wool; and place on the cobble; ~~with your tool~~

~~Human: where's the cobble?~~

~~You: you placed it on the ground~~

~~Human: oh right~~

You: so use your tool and break the gray wool , then place it on the cobblestone

~~Human: how do i pick it up? do i have to break it again~~

You: ~~no;~~ break the gray wool piece over here with your hoe tool

~~Human: how do i switch tools~~

You: ~~uh; numbers;~~ place it on the cobblestone; ~~ah its the other way around; you gotta reverse the order;~~ i can't break them with my tool; ~~nice;~~ so whats the recipe for diamond again? gray wool +

Human: soul sand + gray wool

...

Listing A.2: Example of a filtered dialogue from the Mindcraft dataset

A.2 Baseline Architecture

For easy reference, we provide the architecture used for the CPA tasks in the baseline [BMY+23] in Figure A.2. The architecture which was used by them for the ToM tasks is a subset to the one shown in Figure A.2 but schematically similar.

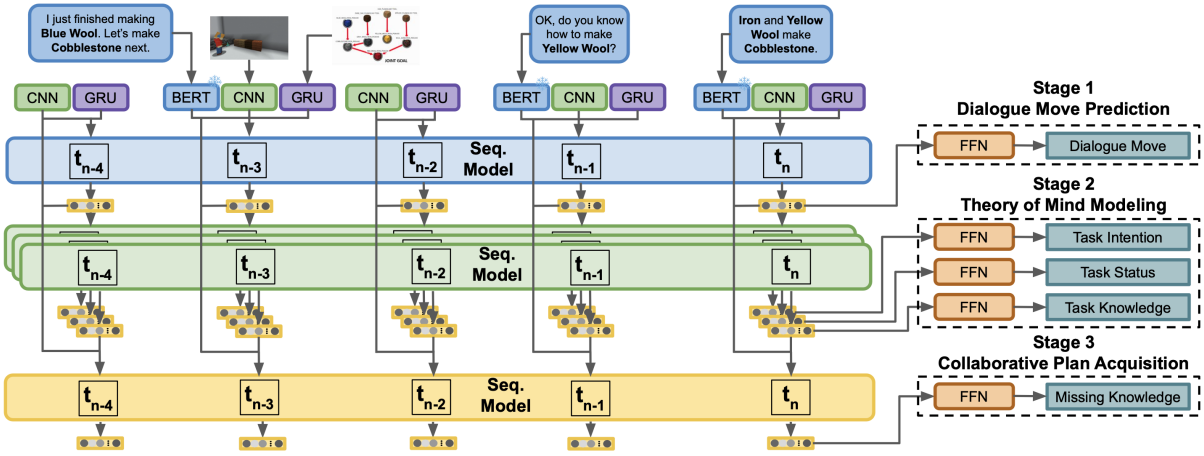

Figure A.2: Architecture used for ToM and CPA tasks in the baseline [BMY+23]

A.3 Dataset Statistics

In order to set the results of our experiments in context, we provide additional statistics about the Mindcraft dataset [BCC21] (see Figures A.3 to A.5).

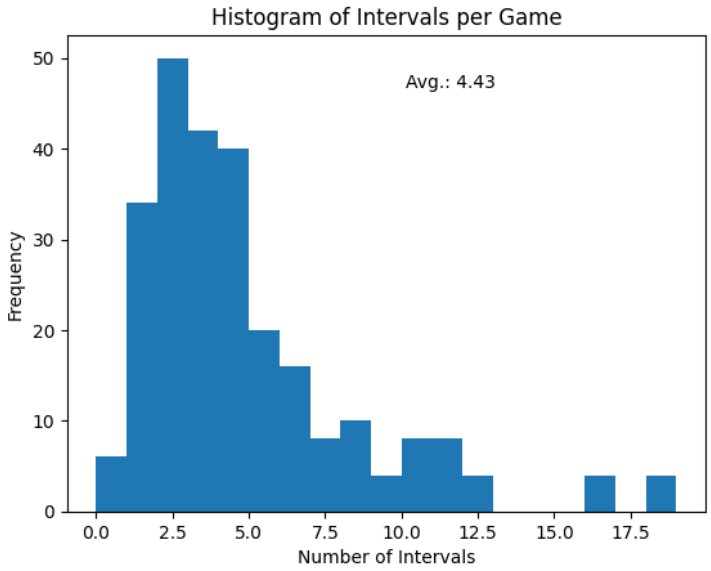

Figure A.3: Intervals per game in the Mindcraft dataset [BMY+23]

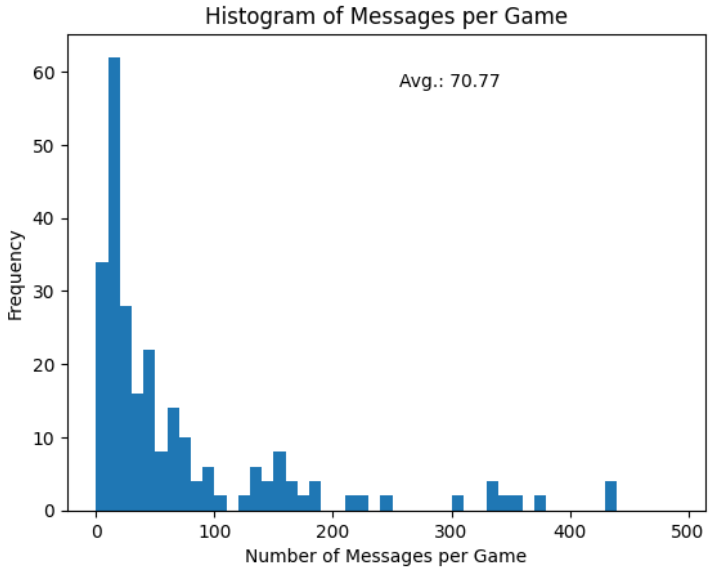

Figure A.4: Messages per game in the Mindcraft dataset [BMY+23]

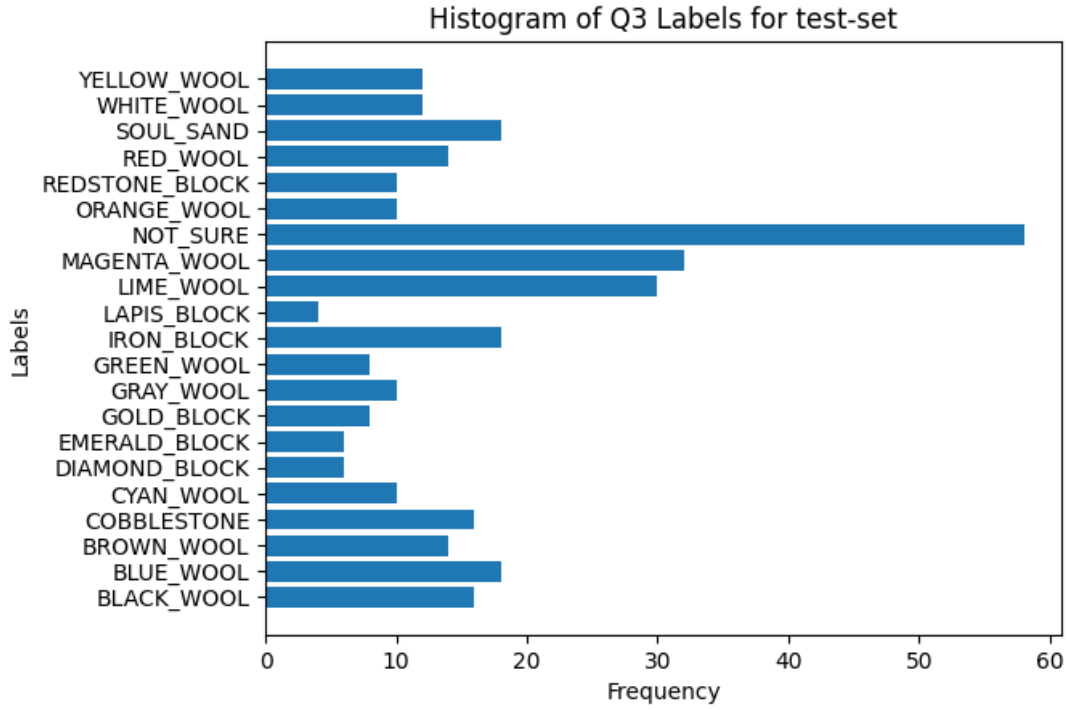

Figure A.5: Label distribution for the task intention ToM task in the Mindcraft dataset [BMY+23]

A.4 Fine-tuning Resource Compromise

In our experiments, we fine-tuned the multimodal model on the ToM and CPA tasks for 10 epochs. However, we also experienced that already one epoch of fine-tuning can lead to good results (see Table A.1).

	Task Status	Task Knowledge	Task Intention
Setup 3, best of 10 epochs	64.8 $\pm$ 0.9	58.8 $\pm$ 0.6	32.6 $\pm$ 2.1
Setup 3, 1 epoch	63.7 $\pm$ 2.8	54.8 $\pm$ 3.2	23.8 $\pm$ 3.5

Table A.1: Comparison of fine-tuning results for 10 epochs and 1 epoch on ToM tasks

A.5 Alternative Hyperparameters

The independent hyperparameter search that we performed has the limitation that dependencies between hyperparameters are not considered. As a result, more optimal

hyperparameter sets might exist. One such alternative hyperparameter set was found later in our work and is presented in Table A.2 and Table A.3. However, we believe that the observations and implications of our work still hold.

Hyperparameter	Value
LoRA r	32
LoRA $alpha$	8
LoRA dropout_rate	**0.3**
learning rate	7×10^{-5}
weight_decay	**0.0001**
projection dropout rate	**0.9**

Table A.2: Alternative hyperparameter set. Differences to the original hyperparameter set are highlighted in bold.

	Task Status	Task Knowledge	Task Intention
Setup 2	70.9	62.5	38.7
Setup 3	67.9	61.0	29.3

Table A.3: F1 scores on the ToM tasks with alternative hyperparameters. Only one run was performed for each setup.

Bibliography

[ADL+22] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, K. Simonyan. *Flamingo: a Visual Language Model for Few-Shot Learning*. 2022. arXiv: 2204.14198 [cs.CV] (cit. on pp. 11, 17).

[ALS+19] A. R. Akula, C. Liu, S. Saba-Sadiya, H. Lu, S. Todorovic, J. Y. Chai, S.-C. Zhu. *X-ToM: Explaining with Theory-of-Mind for Gaining Justified Human Trust*. 2019. arXiv: 1909.06907 [cs.AI] (cit. on p. 14).

[BB12] J. Bergstra, Y. Bengio. "Random Search for Hyper-Parameter Optimization." In: *Journal of Machine Learning Research* 13.10 (2012), pp. 281–305. URL: http://jmlr.org/papers/v13/bergstra12a.html (cit. on p. 58).

[BCC21] C.-P. Bara, S. CH-Wang, J. Chai. "MindCraft: Theory of Mind Modeling for Situated Dialogue in Collaborative Tasks." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.emnlp-main.85. URL: https://doi.org/10.18653%2Fv1%2F2021.emnlp-main.85 (cit. on pp. 12, 13, 18, 21–23, 29, 32, 55, 59, 60, 62).

[BJST17] C. L. Baker, J. Jara-Ettinger, R. Saxe, J. B. Tenenbaum. "Rational quantitative attribution of beliefs, desires and percepts in human mentalizing." In: *Nature Human Behaviour* 1 (2017). URL: https://api.semanticscholar.org/CorpusID:3338320 (cit. on p. 14).

[Blo81] N. Block. "Psychologism and Behaviorism." In: *The Philosophical Review* 90 (Jan. 1981). DOI: 10.2307/2184371 (cit. on p. 15).

[BMR+20] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL] (cit. on p. 15).

[BMY+23] C.-P. Bara, Z. Ma, Y. Yu, J. Shah, J. Chai. "Towards Collaborative Plan Acquisition through Theory of Mind Modeling in Situated Dialogue." In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023. DOI: 10.24963/ijcai.2023/330. URL: https://doi.org/10.24963%2Fijcai.2023%2F330 (cit. on pp. 12, 13, 21, 23, 24, 32, 34, 35, 44, 45, 49, 50, 56, 59, 61–63).

[BPM22] S. H. S. Basha, V. Pulabaigari, S. Mukherjee. "An information-rich sampling technique over spatio-temporal CNN for classification of human actions in videos." In: *Multimedia Tools and Applications* 81.28 (2022), pp. 40431–40449. DOI: 10.1007/s11042-022-12856-6. URL: https://doi.org/10.1007/s11042-022-12856-6 (cit. on p. 47).

[BRA+24] M. Bortoletto, C. Ruhdorfer, A. Abdessaied, L. Shi, A. Bulling. "Limits of Theory of Mind Modelling in Dialogue-Based Collaborative Plan Acquisition." In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2024, pp. 1–16. arXiv: 2405.12621 [cs.AI] (cit. on pp. 13, 32, 44, 45, 49, 50).

[BRSB24] M. Bortoletto, C. Ruhdorfer, L. Shi, A. Bulling. "Benchmarking Mental State Representations in Language Models." In: *arXiv preprint* (2024) (cit. on pp. 12, 50).

[BS11] C. Baker, R. Saxe. "Bayesian Theory of Mind: Modeling Joint Belief-Desire Attribution." In: *Proceedings of the Thirty-Third Annual Conference of the Cognitive Science Society* (Jan. 2011) (cit. on p. 14).

[BSB24] M. Bortoletto, L. Shi, A. Bulling. "Neural Reasoning About Agents' Goals, Preferences, and Actions." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 1. 2024, pp. 456–464 (cit. on p. 14).

[CHL+22] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, J. Wei. *Scaling Instruction-*

Finetuned Language Models. 2022. arXiv: 2210.11416 [cs.LG] (cit. on p. 16).

[CKH13] S. M. Carlson, M. A. Koenig, M. B. Harms. "Theory of mind." In: *WIREs Cognitive Science* 4.4 (2013), pp. 391–402. DOI: https://doi.org/10.1002/wcs.1232. eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcs.1232. URL: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcs.1232 (cit. on p. 11).

[CLL+23] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, E. P. Xing. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. Mar. 2023. URL: https://lmsys.org/blog/2023-03-30-vicuna/ (cit. on pp. 11, 16).

[CWZ+24] Z. Chen, J. Wu, J. Zhou, B. Wen, G. Bi, G. Jiang, Y. Cao, M. Hu, Y. Lai, Z. Xiong, M. Huang. *ToMBench: Benchmarking Theory of Mind in Large Language Models*. 2024. arXiv: 2402.15052 [cs.CL] (cit. on p. 15).

[CXZG16] T. Chen, B. Xu, C. Zhang, C. Guestrin. *Training Deep Nets with Sublinear Memory Cost*. 2016. arXiv: 1604.06174 [cs.LG] (cit. on pp. 19, 20, 29).

[DFE+22] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, C. Ré. *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness*. 2022. arXiv: 2205.14135 [cs.LG] (cit. on p. 20).

[DLD+23] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, Z. Sui. *A Survey on In-context Learning*. 2023. arXiv: 2301.00234 [cs.CL] (cit. on p. 19).

[DMD+23] J. Ding, S. Ma, L. Dong, X. Zhang, S. Huang, W. Wang, N. Zheng, F. Wei. *LongNet: Scaling Transformers to 1,000,000,000 Tokens*. 2023. arXiv: 2307.02486 [cs.CL] (cit. on p. 20).

[DPHZ23] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer. *QLoRA: Efficient Finetuning of Quantized LLMs*. 2023. arXiv: 2305.14314 [cs.LG] (cit. on pp. 19, 26).

[ENO+21] N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, C. Olah. "A Mathematical Framework for Transformer Circuits." In: *Transformer Circuits Thread* (2021). https://transformer-circuits.pub/2021/framework/index.html (cit. on pp. 15, 16).

[GEL+23] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, I. Misra. *ImageBind: One Embedding Space To Bind Them All*. 2023. arXiv: 2305.05665 [cs.CV] (cit. on pp. 25, 46).

[GNP+23] W. Gurnee, N. Nanda, M. Pauly, K. Harvey, D. Troitskii, D. Bertsimas. *Finding Neurons in a Haystack: Case Studies with Sparse Probing*. 2023. arXiv: 2305.01610 [cs.LG] (cit. on p. 16).

[GW92] A. Gopnik, H. M. Wellman. "Why the Child's Theory of Mind Really Is a Theory." In: *Mind & Language* 7.1-2 (1992), pp. 145–171. DOI: https://doi.org/10.1111/j.1468-0017.1992.tb00202.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1468-0017.1992.tb00202.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0017.1992.tb00202.x (cit. on p. 11).

[HS44] F. Heider, M. Simmel. "An Experimental Study of Apparent Behavior." In: *The American Journal of Psychology* 57.2 (1944), pp. 243–259. ISSN: 00029556. URL: http://www.jstor.org/stable/1416950 (visited on 11/06/2023) (cit. on p. 14).

[HS97] S. Hochreiter, J. Schmidhuber. "Long Short-term Memory." In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735 (cit. on p. 24).

[HSW+21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.CL] (cit. on pp. 19, 26, 31, 41).

[HWL+23] Z. Hu, L. Wang, Y. Lan, W. Xu, E.-P. Lim, L. Bing, X. Xu, S. Poria, R. K.-W. Lee. *LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models*. 2023. arXiv: 2304.01933 [cs.CL] (cit. on p. 19).

[JNH20] P. Jayannavar, A. Narayan-Chen, J. Hockenmaier. "Learning to execute instructions in a Minecraft dialogue." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by D. Jurafsky, J. Chai, N. Schluter, J. Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 2589–2602. DOI: 10.18653/v1/2020.acl-main.232. URL: https://aclanthology.org/2020.acl-main.232 (cit. on p. 18).

[JSM+23] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL] (cit. on pp. 16, 26, 58).

[JWC+23] C. Jin, Y. Wu, J. Cao, J. Xiang, Y.-L. Kuo, Z. Hu, T. Ullman, A. Torralba, J. Tenenbaum, T. Shu. "MMToM-QA: Multimodal Theory of Mind Question Answering." In: *Submitted to The Twelfth International Conference on Learning Representations*. under review. 2023. URL: https://openreview.net/forum?id=sMFqEror1b (cit. on p. 17).

[KB17] D. P. Kingma, J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG] (cit. on p. 19).

[KGR+23] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa. *Large Language Models are Zero-Shot Reasoners*. 2023. arXiv: 2205.11916 [cs.CL] (cit. on p. 15).

[KLJ+07] M. Krych-Appelbaum, J. B. Law, D. Jones, A. Barnacz, A. Johnson, J. P. Keenan. ""I think I know what you mean": The role of theory of mind in collaborative communication." In: *Interaction Studies* 8.2 (2007), pp. 267–280. ISSN: 1572-0373. DOI: https://doi.org/10.1075/is.8.2.05kry. URL: https://www.jbe-platform.com/content/journals/10.1075/is.8.2.05kry (cit. on p. 17).

[KSZ+23] H. Kim, M. Sclar, X. Zhou, R. L. Bras, G. Kim, Y. Choi, M. Sap. *FANToM: A Benchmark for Stress-testing Machine Theory of Mind in Interactions*. 2023. arXiv: 2310.15421 [cs.CL] (cit. on p. 15).

[KWH22] F. D. Keles, P. M. Wijewardena, C. Hegde. *On The Computational Complexity of Self-Attention*. 2022. arXiv: 2209.04881 [cs.LG] (cit. on pp. 20, 28).

[LCS+23] H. Li, Y. Q. Chong, S. Stepputtis, J. Campbell, D. Hughes, M. Lewis, K. Sycara. *Theory of Mind for Multi-Agent Collaboration via Large Language Models*. 2023. arXiv: 2310.10701 [cs.CL] (cit. on pp. 15, 17, 18).

[LHH+24] Y. Liu, H. He, T. Han, X. Zhang, M. Liu, J. Tian, Y. Zhang, J. Wang, X. Gao, T. Zhong, Y. Pan, S. Xu, Z. Wu, Z. Liu, X. Zhang, S. Zhang, X. Hu, T. Zhang, N. Qiang, T. Liu, B. Ge. *Understanding LLMs: A Comprehensive Overview from Training to Inference*. 2024. arXiv: 2401.02038 [cs.CL] (cit. on p. 29).

[LHW+23] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, Y. Qiao. *VideoChat: Chat-Centric Video Understanding*. 2023. arXiv: 2305.06355 [cs.CV] (cit. on pp. 11, 17).

[LLSH23] J. Li, D. Li, S. Savarese, S. Hoi. *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. 2023. arXiv: 2301.12597 [cs.CV] (cit. on pp. 11, 17).

[LLWL23] H. Liu, C. Li, Q. Wu, Y. J. Lee. *Visual Instruction Tuning*. 2023. arXiv: 2304.08485 [cs.CV] (cit. on pp. 17, 32).

[LTAE24] J. Lin, N. Tomlin, J. Andreas, J. Eisner. *Decision-Oriented Dialogue for Human-AI Collaboration*. 2024. arXiv: 2305.20076 [cs.CL] (cit. on p. 17).

[LTV23] C. Leer, V. Trost, V. Voruganti. *Violation of Expectation via Metacognitive Prompting Reduces Theory of Mind Prediction Error in Large Language Models*. 2023. arXiv: 2310.06983 [cs.CL] (cit. on p. 15).

[MFS+02] P. McGuire, J. Fritsch, J. Steil, F. Rothling, G. Fink, S. Wachsmuth, G. Sagerer, H. Ritter. "Multi-modal human-machine communication for instructing robot grasping tasks." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 2. 2002, 1082–1088 vol.2. DOI: 10.1109/IRDS.2002.1043875 (cit. on p. 18).

[MH23] S. R. Moghaddam, C. J. Honey. *Boosting Theory-of-Mind Performance in Large Language Models via Prompting*. 2023. arXiv: 2304.11490 [cs.AI] (cit. on pp. 15, 19).

[MLZ+23] Y. Mao, S. Liu, P. Zhao, Q. Ni, X. Lin, L. He. *A Review on Machine Theory of Mind*. 2023. arXiv: 2303.11594 [cs.AI] (cit. on pp. 11, 15).

[MSPC23] Z. Ma, J. Sansom, R. Peng, J. Chai. *Towards A Holistic Landscape of Situated Theory of Mind in Large Language Models*. 2023. arXiv: 2310.19619 [cs.CL] (cit. on p. 15).

[MWM+24] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, F. Wei. *The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits*. 2024. arXiv: 2402.17764 [cs.CL] (cit. on p. 20).

[NBM19] S. Narang, A. Best, D. Manocha. "Inferring User Intent using Bayesian Theory of Mind in Shared Avatar-Agent Virtual Environments." In: *IEEE Transactions on Visualization and Computer Graphics* 25.5 (2019), pp. 2113–2122. DOI: 10.1109/TVCG.2019.2898800 (cit. on pp. 14, 15).

[NJH19] A. Narayan-Chen, P. Jayannavar, J. Hockenmaier. "Collaborative Dialogue in Minecraft." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by A. Korhonen, D. Traum, L. Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5405–5415. DOI: 10.18653/v1/P19-1537. URL: https://aclanthology.org/P19-1537 (cit. on p. 18).

[NNL+22] D. Nguyen, P. Nguyen, H. Le, K. Do, S. Venkatesh, T. Tran. *Learning Theory of Mind via Dynamic Traits Attribution*. 2022. arXiv: 2204.09047 [cs.LG] (cit. on p. 14).

[NNL+23] D. Nguyen, P. Nguyen, H. Le, K. Do, S. Venkatesh, T. Tran. *Memory-Augmented Theory of Mind Network*. 2023. arXiv: 2301.06926 [cs.AI] (cit. on p. 12).

[OCSS23] I. Oguntola, J. Campbell, S. Stepputtis, K. Sycara. *Theory of Mind as Intrinsic Motivation for Multi-Agent Reinforcement Learning*. 2023. arXiv: 2307.01158 [cs.LG] (cit. on pp. 14, 15).

[OHS21] I. Oguntola, D. Hughes, K. Sycara. *Deep Interpretable Models of Theory of Mind*. 2021. arXiv: 2104.02938 [cs.LG] (cit. on p. 14).

[Ope23] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL] (cit. on p. 16).

[OWJ+22] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe. *Training language models to follow instructions with human feedback*. 2022. arXiv: 2203.02155 [cs.CL] (cit. on p. 16).

[PW78] D. Premack, G. Woodruff. "Does the chimpanzee have a theory of mind?" In: *Behavioral and Brain Sciences* 1.4 (1978), pp. 515–526. DOI: 10.1017/S0140525X00076512 (cit. on pp. 11, 14).

[RPS+18] N. C. Rabinowitz, F. Perbet, H. F. Song, C. Zhang, S. M. A. Eslami, M. Botvinick. *Machine Theory of Mind*. 2018. arXiv: 1802.07740 [cs.AI] (cit. on pp. 11, 12, 14).

[SCTS23] M. Sclar, Y. Choi, Y. Tsvetkov, A. Suhr. *Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting*. 2023. arXiv: 2310.11324 [cs.CL] (cit. on pp. 19, 57).

[SKW+23] M. Sclar, S. Kumar, P. West, A. Suhr, Y. Choi, Y. Tsvetkov. *Minding Language Models' (Lack of) Theory of Mind: A Plug-and-Play Multi-Character Belief Tracker*. 2023. arXiv: 2306.00924 [cs.CL] (cit. on pp. 14, 15).

[SLFC23] M. Sap, R. LeBras, D. Fried, Y. Choi. *Neural Theory-of-Mind? On the Limits of Social Intelligence in Large LMs*. 2023. arXiv: 2210.13312 [cs.CL] (cit. on p. 15).

[SLL+23] Y. Su, T. Lan, H. Li, J. Xu, Y. Wang, D. Cai. *PandaGPT: One Model To Instruction-Follow Them All*. 2023. arXiv: 2305.16355 [cs.CL] (cit. on p. 17).

[SOW+22] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, P. Christiano. *Learning to summarize from human feedback*. 2022. arXiv: 2009.01325 [cs.CL] (cit. on p. 16).

[SYS+22] A. Suhr, C. Yan, C. Schluger, S. Yu, H. Khader, M. Mouallem, I. Zhang, Y. Artzi. *Executing Instructions in Situated Collaborative Interactions*. 2022. arXiv: 1910.03655 [cs.CL] (cit. on p. 18).

[TGZ+23] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto. *Stanford Alpaca: An Instruction-following LLaMA model*. https://github.com/tatsu-lab/stanford_alpaca. 2023 (cit. on pp. 11, 16).

[TLI+23] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL] (cit. on pp. 11, 16).

[TMS+23] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL] (cit. on p. 11).

[UA19] T. Udagawa, A. Aizawa. *A Natural Language Corpus of Common Grounding under Continuous and Partially-Observable Context*. 2019. arXiv: 1907.03399 [cs.CL] (cit. on p. 18).

[Ull23] T. Ullman. *Large Language Models Fail on Trivial Alterations to Theory-of-Mind Tasks*. 2023. arXiv: 2302.08399 [cs.AI] (cit. on pp. 15, 57).

[VSP+23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL] (cit. on pp. 15, 24).

[WFH+23] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, D. C. Schmidt. *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. 2023. arXiv: 2302.11382 [cs.SE] (cit. on p. 19).

[WFQ+23] S. Wu, H. Fei, L. Qu, W. Ji, T.-S. Chua. *NExT-GPT: Any-to-Any Multimodal LLM*. 2023. arXiv: 2309.05519 [cs.AI] (cit. on pp. 11, 12, 17, 24, 25, 27, 32, 43).

[WWE+20] R. E. Wang, S. A. Wu, J. A. Evans, J. B. Tenenbaum, D. C. Parkes, M. Kleiman-Weiner. "Too Many Cooks: Coordinating Multi-Agent Collaboration Through Inverse Planning." In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*.

AAMAS '20. Auckland, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems, 2020, pp. 2032–2034. ISBN: 9781450375184 (cit. on p. 15).

[WWS+23] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL] (cit. on p. 15).

[XXQ+23] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, F. L. Wang. *Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment*. 2023. arXiv: 2312.12148 [cs.CL] (cit. on p. 19).

[YDF08] W. Yoshida, R. J. Dolan, K. J. Friston. "Game Theory of Mind." In: *PLOS Computational Biology* 4.12 (Dec. 2008), pp. 1–14. DOI: 10.1371/journal.pcbi.1000254. URL: https://doi.org/10.1371/journal.pcbi.1000254 (cit. on p. 14).

[YFZ+23] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, E. Chen. *A Survey on Multimodal Large Language Models*. 2023. arXiv: 2306.13549 [cs.CV] (cit. on p. 16).

[ZGP+22] T. Zhi-Xuan, N. Gothoskar, F. Pollok, D. Gutfreund, J. B. Tenenbaum, V. K. Mansinghka. *Solving the Baby Intuitions Benchmark with a Hierarchically Bayesian Theory of Mind*. 2022. arXiv: 2208.02914 [cs.AI] (cit. on p. 15).

[ZLB23] H. Zhang, X. Li, L. Bing. *Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding*. 2023. arXiv: 2306.02858 [cs.CL] (cit. on pp. 11, 17, 32).

[ZLZ+23] D. Zhang, S. Li, X. Zhang, J. Zhan, P. Wang, Y. Zhou, X. Qiu. *SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities*. 2023. arXiv: 2305.11000 [cs.CL] (cit. on p. 17).

[ZNB21] H. Zhu, G. Neubig, Y. Bisk. *Few-shot Language Coordination by Modeling Theory of Mind*. 2021. arXiv: 2107.05697 [cs.CL] (cit. on p. 36).

[ZZW24] W. Zhu, Z. Zhang, Y. Wang. *Language Models Represent Beliefs of Self and Others*. 2024. arXiv: 2402.18496 [cs.AI] (cit. on pp. 11, 12, 15, 50, 52).

All links were last followed on June 9th, 2024.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature