PersuHMM: Iterative Learning the Hierarchical Meta-Strategy Memory for Persuasive Dialogue

Yanyue Zhang[1] , **Zihao Wang**[1] , **Xin Zhang**[2] and **Deyu Zhou**[1]
[1] School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China
[2] Department of Computer Science, The University of Manchester, United Kingdom
{yanyuez98, 220252256}@seu.edu.cn, xin.zhang-2@manchester.ac.uk, d.zhou@seu.edu.cn,

Abstract

Persuasive dialogue aims to alter people's attitudes or behaviors through conversation . While LLMs can generate emotional responses, they tend to use a uniform approach across varied scenarios, limiting their persuasive effectiveness. Diverse demands need varied persuasion strategies, which challenge both LLMs and human efforts. To address this, this paper incorporates the idea of meta-learning and proposes PersuHMM, an iterative learning approach for constructing the **H**ierarchical **M**eta-strategy **M**emory to generate **Persua**sive dialogue. Through the interaction between Meta-layer and Task-layer, the approach continuously accumulates automated meta-strategies from generated outcomes and data labels, forming a persuasive meta-strategy memory that balances efficiency and generalizability. To facilitate model retrieval and use, the meta-strategy memory is hierarchically organized via clustering for a clearer and more logical storage structure. Experimental results show that the hierarchical meta-strategy memory enhances both the persuasiveness and generalizability of the persuasion model, and even enables cross-model performance improvement.

1 Introduction

Persuasive dialogue, defined as communication intended to shape or change a recipient's responses, has been extensively studied for decades [Stiff and Mongeau, 2016; Dönmez and Falenska, 2025; Salvi *et al.*, 2024]. In fields such as politics [Potter *et al.*, 2024; Bai *et al.*, 2025], public health [Costello *et al.*, 2024; Altay *et al.*, 2023], and charitable donations [Mishra *et al.*, 2022], large language model (LLM)-based persuasive text generation has achieved or even surpassed human-level persuasiveness [Rogiers *et al.*, 2024].

While mainstream persuasive dialogue datasets cover diverse domains, experimental results reveal significant performance variation across them. As shown in Figure 1, neither instruction-based nor fine-tuning-based methods exceed a 55% Win Rate, with notable fluctuations reflecting the lack of adaptive mechanisms for diverse persuasion scenarios. Underlying this variation are distinct persuasive needs across

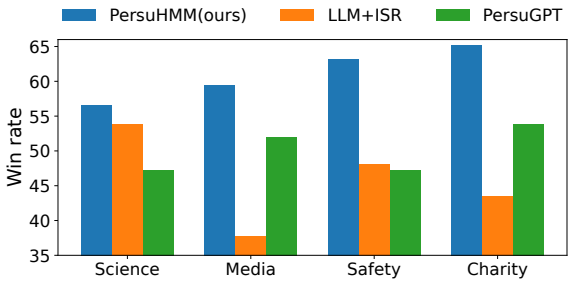

Figure 1: Performance comparison of persuasion methods across domains. PersuGPT is implemented by fine-tuning an LLM. LLM+ISR is implemented by guiding the LLM to perform intent-to-strategy reasoning.

fields, necessitating differentiated strategies [Hackenburg *et al.*, 2024]. Yet, merely instructing LLMs to analyze persuasion intent and strategies (LLM+ISR) still results in clear domain gaps: performance remains stronger in familiar areas like science but drops sharply in domains such as media.

Relying solely on LLMs for strategy reasoning is constrained by their inherent capabilities. Conversely, depending on human experts to provide such strategies is costly and may not be applicable for LLMs. To provide strategic guidance for LLMs in persuasion, we leverage the idea of meta-learning to automate the acquisition of meta-strategy memory through interaction between Meta-layer and Task-layer. Unlike traditional meta-learning that focuses on model parameters, our approach facilitates iterative, automated accumulation of strategic knowledge, resulting in a meta-strategy memory that balances specificity and generalizability.

Additionally, the memory acquired from real-world scenarios contains various types of meta-strategies, whose similarities and differences can impede the model's comprehension. Therefore, we employ clustering-based hierarchical memory management to facilitate the model's understanding of inter-strategy relationships and enable more targeted applications.

Ultimately, we propose PersuHMM, an iterative learning approach that builds a Hierarchical Meta-strategy Memory to generate Persuasive dialogues. The memory acquisition process is initiated through the interaction between two layers: Meta-layer and Task-layer. The Meta-layer, equipped with a strategy generator, a strategy refiner, and a step genera-

tor, is responsible for selecting meta-strategies and producing step-by-step guidance for the task-layer. It subsequently updates the strategy memory using batches of success patterns accumulated by the task-layer. The task-layer, comprising a persuasion model, a goal evaluator, and a pattern analyzer, executes the current persuasion task and analyzes the outcomes. It extracts successful experience from positive results or labeled data and feeds this knowledge back to the Meta-layer, enabling refinement of persuasion strategies.

After completing iterative meta-strategy learning, hierarchical memory management is implemented to enhance model usability. This process leverages domain information, strategy types, and K-means clustering results, yielding a structured hierarchical meta-strategy memory. Subsequently, the most relevant meta-strategies for the current persuasion context are retrieved from memory to generate targeted persuasion steps and goals, which then guide the persuasion model in generating context-specific dialogue.

The contributions of this paper are as follows:

- Inspired by meta-learning, we propose a persuasive dialogue approach guided by meta-strategy memory, which automates the acquisition of high-quality persuasion strategies balancing effectiveness and generalizability through interaction between Meta-layer and Task-layer.
- We hierarchically structure the meta-strategy memory via domain information, strategy types, and K-means clustering to enhance model interpretability and utilization.
- Experiments demonstrate that the hierarchical meta-strategy memory improves the model's persuasive ability and exhibits both domain and model generalization capabilities.

2 Related Work

Persuasive dialogue has been a long-standing area of research. Since the emergence of large language models (LLMs), some studies have tested their capabilities compared with human [Dönmez and Falenska, 2025] or in specific persuasive tasks such as charity donations [Mishra *et al.*, 2022], public health [Costello *et al.*, 2024; Altay *et al.*, 2023], debate [Salvi *et al.*, 2024], and politics [Potter *et al.*, 2024; Bai *et al.*, 2025]. Other works have tested the ability of LLMs to detect the persuader's intent and express illocutionary intent during the persuasion process [Sakurai and Miyao, 2024]. Some research focuses on designing evaluation systems for persuasive dialogues, and assessing their effects by predicting the persuasiveness of new arguments [Saenger *et al.*, 2024]. In terms of empathy, recent works focus on enhancing the emotional comprehension capability of LLM responses through prompt engineering [Wu *et al.*, 2025] and fine-tuning techniques [Zhang *et al.*, 2025; Xie *et al.*, 2025], or decouple strategy prediction from language generation to address LLMs' inherent preference bias towards certain emotional strategies [Wan *et al.*, 2025]. There are also works that focus on using manually designed or off-the-shelf persuasion strategies to guide persuasive dialogues: PC-CRS uses social psychology-based manually designed strategies to guide a movie recommendation system [Qin *et al.*, 2024]. SARA designs an auto-regressive predictor to match the semantics and strategies from PersuasionForGood [Jin *et al.*, 2023]. Some studies utilize LLM prompt engineering to learn persuasive strategies: DebateQD uses LLMs to generate new strategies based on existing strategies and continuously eliminates ineffective strategies through LLM debates [Reedi *et al.*, 2025]. However, these strategy-based methods are primarily based on persuasion or debate data from a single domain and do not consider the specificity or generalization of strategies across different domains. Still other works improve the persuasive ability of language models through fine-tuning methods: Specific reward functions are used to improve the model's response in specific aspects, such as empathy [Samad *et al.*, 2022] and politeness [Mishra *et al.*, 2022]. PersuGPT fine-tunes a persuasion model for multiple domains through intent-to-strategy reasoning [Jin *et al.*, 2024]. These fine-tuning methods are often constrained by the performance limits of the base models.

3 Methodology

In this section, we provide a detailed description of the generation process of persuasive dialogue via PersuHMM and the acquisition of the hierarchical meta-strategy memory.

3.1 Persuasion via PersuHMM

Given a persuasion task $\boldsymbol{x_i}$, the objective of PersuHMM is to engage the persuadee and gradually shift their stance through a multi-turn dialogue $\boldsymbol{Y_i}$. The generation process of persuasive dialogue is illustrated in Figure 2, including three steps.

Hierarchical Meta-Strategy Memory Retrieval

As illustrated in Figure 2, the meta-strategy memory is represented as a rooted tree. In the tree, each leaf node corresponds to a fine-grained persuasion strategy, and strategies are first grouped by persuasion domains (Culture, Media, Science, etc.) and then further divided into three categories: logical, emotional, and behavioral strategies.

For task $\boldsymbol{x_i}$, we first automatically obtain a strategy subtree $\boldsymbol{S_i} = \bigcup_j \boldsymbol{S^{meta}}[\boldsymbol{d_j}]$, where $\boldsymbol{S^{meta}}$ denotes the meta-strategy memory, $\boldsymbol{d_j}$ denotes a persuasion domain of task $\boldsymbol{x_i}$. Then the strategy subtree needs to be adjusted when the following two cases arise: (1) Persuasion domains with limited data may result in insufficient strategy learning. For tasks within such domains, we further include strategies from its sibling domains into $\boldsymbol{S_i}$ to compensate for data sparsity. (2) When the context length is limited (GPT-3.5-turbo, etc.), the LLM may not be able to input the full subset $\boldsymbol{S_i}$. For such cases, we compute the sentence embedding vector similarity between task $\boldsymbol{x_i}$ and each strategy in subset $\boldsymbol{S_i}$. Then we select the top-k strategies ranked by similarity. For selected strategies with sibling nodes under the same macro-strategy, we extract the subtree rooted at the macro-strategy. After extraction, the strategies are reorganized into a tree structure.

Strategic Step Generation

Based on the task $\boldsymbol{x_i}$ and the subtree $\boldsymbol{S_i}$, an LLM Step Generator generate a persuasion plan $\boldsymbol{P_i}$. Plan $\boldsymbol{P_i}$ includes a sequence of paired persuasion steps and stage goals, with each

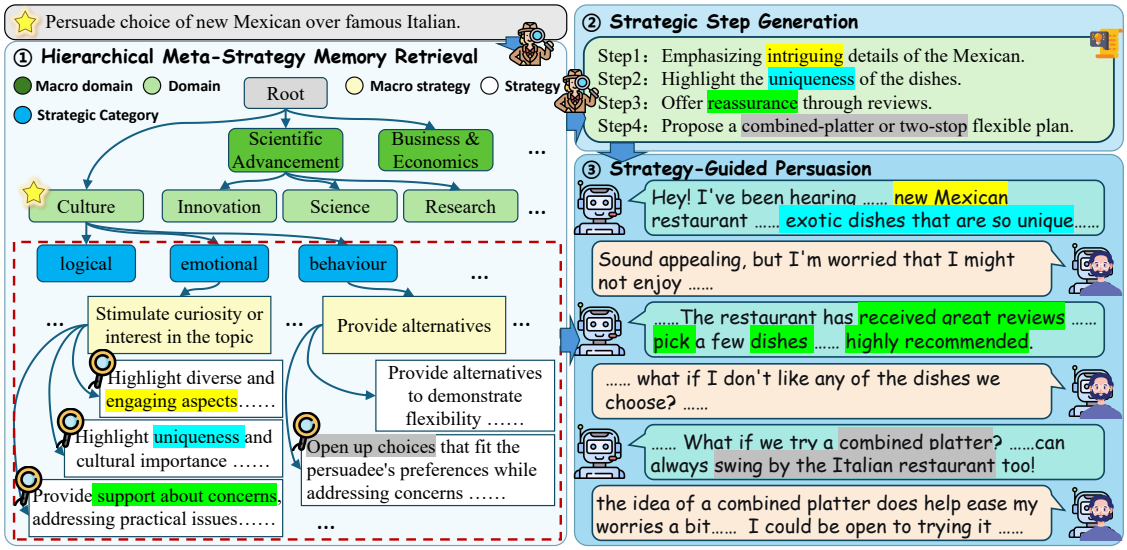

Figure 2: The generation process of persuasive dialogue via PersuHMM.

pair corresponding to one strategy. The persuasion steps reflect how the persuadee is engaged, guided toward the goal, and gradually convinced to take action, and the stage goal is defined as changes in the persuadee's state, emotion, viewpoint, or behavior. In Figure 2, each step and its corresponding strategy are highlighted in the same color.

Strategy-Guided Persuasion

We use the plan $\boldsymbol{P_i}$ and subtree $\boldsymbol{S_i}$ to guide the persuader to conduct a multi-turn dialogue with the persuadee, until the persuasion task is completed or the persuadee voluntarily ends the dialogue. Each step and its corresponding dialogue pieces are highlighted in the same color in Figure 2.

3.2 Hierarchical Meta-Strategy Memory Acquisition

As illustrated in Figure 3, the hierarchical meta-strategy memory acquisition comprises two steps: iterative meta-strategy learning and clustering-based hierarchization.

Iterative Meta-Strategy Learning

To build a meta-strategy memory with efficiency and generalization, we introduce an iterative learning cycle between Meta-layer and Task-layer. As shown in Figure 3, Task-layer executes persuasion under the guidance of the Meta-layer and accumulates success patterns. The Meta-layer provides a persuasion plan to Task-layer based on the current meta-strategy memory, refines new strategies from the success patterns and updates the meta-strategy memory.

In the learning process, we iteratively execute the following workflow: (1) We sample a persuasion task$\boldsymbol{x_i}$ from the dataset and obtain a strategy subtree $\boldsymbol{S_i}$ corresponding to its domains. (2) Based on the subtree and the task, **Step Generator** outputs a persuasion plan $\boldsymbol{P_i}$ including a sequence of paired persuasion steps and stage goals, with corresponding meta-strategies. (3) A **persuader** model conducts a multi-turn persuasive dialogue $\hat{\boldsymbol{Y}}_i$ with the **persuadee** following the generated plan. (4) The **Goal Evaluator** scores the completion of each stage goal in the dialogue, and (5) The **Pattern Analyst** summarizes a series of success patterns based on these scores, our dialogue $\hat{\boldsymbol{Y}}_i$ and the ground-truth $\boldsymbol{Y_i}$ dialogue in the dataset. (6) After accumulating a sufficient number of success patterns, Task-layer feeds these patterns back to the Meta-layer, where the **Strategy Generator** generates new strategies. (7) The **Strategy Refiner** integrates these new strategies into the meta-strategy memory, which is then used to further guide Task-layer.

The detailed design of each module is as follows.

Step Generator The Step Generator formulates a persuasion plan $\boldsymbol{P_i}$ for Task-layer with appropriate strategies to guide persuasion. As mentioned in 3.1, we first obtain a strategy subtree $\boldsymbol{S_i}$. Then we prompt the LLM to take the task $\boldsymbol{x_i}$ and the subtree $\boldsymbol{S_i}$ as input, and outputs a persuasion plan $\boldsymbol{P_i} = [(\boldsymbol{u_{ij}}, \boldsymbol{g_{ij}}, \boldsymbol{s_{ij}})]_{j=1}^{K}$, where $\boldsymbol{u_{ij}}$ denotes the j-th persuasion step, $\boldsymbol{g_{ij}}$ denotes the stage goal of step $\boldsymbol{u_{ij}}$, and $\boldsymbol{s_{ij}}$ denotes the strategy applied at step $\boldsymbol{u_{ij}}$. We require that the output explicitly indicate the correspondence between steps, goals, and strategies to clarify which strategies the persuader employs to achieve each goal.

Persuader and Persuadee We prompt an LLM to act as the persuader in task $\boldsymbol{x_i}$. Under the guidance of plan $\boldsymbol{P_i}$ and the strategy subtree $\boldsymbol{S_i}$, LLM persuader conducts a multi-turn persuasive dialogue with an LLM persuadee. Since the persuadee's responses are unpredictable, persuasion may deviate from the predefined plan. Thus, while the persuader is expected to follow the plan $\boldsymbol{P_i}$, it is also allowed to employ strategies from the strategy subtree $\boldsymbol{S_i}$ when unexpected situations arise. We finally obtain a persuasive dialogue $\hat{\boldsymbol{Y}}_i$.

Goal Evaluator Goal Evaluator assesses the completion degree of each goal $\boldsymbol{g_{ij}}$ in dialogue $\hat{\boldsymbol{Y}}_i$. Goal Evaluator takes the plan $\boldsymbol{P_i}$ and the dialogue $\hat{\boldsymbol{Y}}_i$ as input, and outputs a score list $\boldsymbol{R_i} = [\boldsymbol{r_{i1}}, \boldsymbol{r_{i2}}, \ldots, \boldsymbol{r_{iK}}]$, where $\boldsymbol{r_{ij}}$ is an integer from

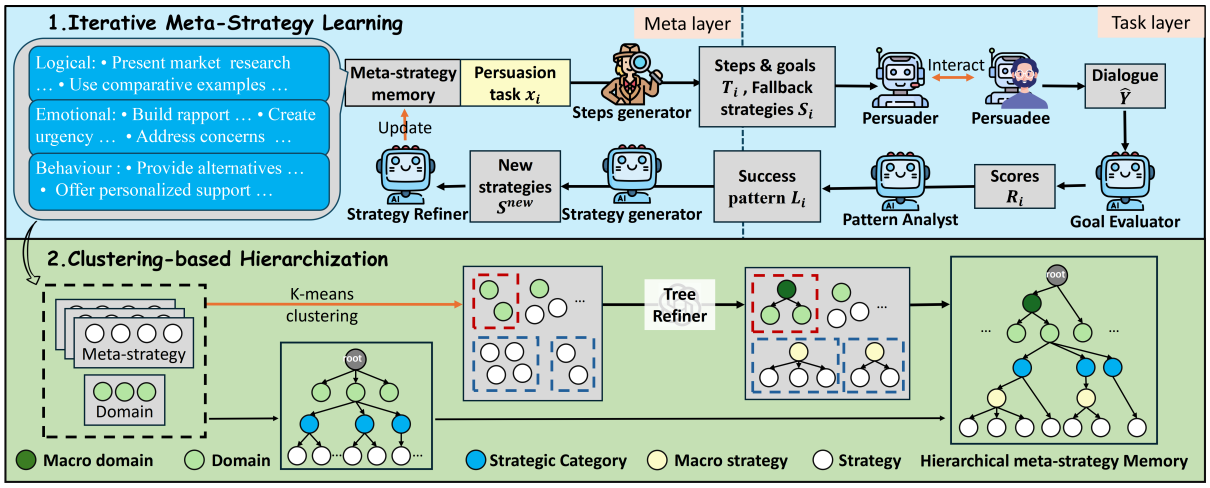

Figure 3: The process of constructing the hierarchical meta-strategy memory.

1 to 5, corresponding to the completion score of g_{ij}. To ensure reliable evaluation, we prompt the LLM to assign scores based on explicit changes in the persuadee's attitude, actions taken, or commitments made.

Pattern Analyst With the completion scores, Pattern Analyst summarizes success patterns in dialogues, facilitating the Meta-layer to evaluate existing strategies and refine new strategies. A success pattern is defined as a concise description including the persuasion domain, the background, the persuader's action, the persuadee's response, adaptive follow-up actions, and the final achieved goal.

Before detailing the Pattern Analyst, two key components per strategy must be introduced: the **validity value** and the **success pattern list**, which are maintained to evaluate strategy quality and record success patterns.

The **validity value** of a strategy is defined as $V = n_{succ}/n_d$, where n_{succ} denotes the success count of a strategy, and n_d denotes the number of times the strategy's domain appears in the learning process after it is generated. Compared with the success rate, validity focuses more on the generalization of strategies, ensuring that only strategies that can achieve success in diverse tasks will get high ratings.

The **success pattern list** of a strategy is initialized with the patterns from which the strategy is abstracted, and is continuously updated with new patterns observed during the learning process. We retain W success patterns with the largest difference in sentence embedding vectors to ensure diversity.

Pattern Analyst outputs two forms of success pattern lists simultaneously. It can be represented as:

$$Analyst(\boldsymbol{Y_i}, \hat{\boldsymbol{Y}}_i, \boldsymbol{R_i}, \boldsymbol{P_i}) = \{\boldsymbol{L_{i1}}, \boldsymbol{L_{i2}}\} \quad (1)$$

where $\boldsymbol{Y_i}$ is the ground-truth dialogue of the task $\boldsymbol{x_i}$. (1) For stage goals that are well accomplished in dialogue $\hat{\boldsymbol{Y}}_i$, the form of the pattern list is $\boldsymbol{L_{i1}} = [(\boldsymbol{s_{ij}}, \boldsymbol{p_{ij}})]_{j=1}^{N}$, where $\boldsymbol{s_{ij}}$ is the strategy employed to achieve pattern $\boldsymbol{p_{ij}}$. $\boldsymbol{L_{i1}}$ is immediately used to update the validity value and success pattern list of $\boldsymbol{s_{ij}}$. (2) For stage goals that fail in $\hat{\boldsymbol{Y}}_i$ but are well accomplished in $\boldsymbol{Y_i}$, or stage goals that are absent in $\boldsymbol{P_i}$ but positively contribute to the persuasion in $\hat{\boldsymbol{Y}}_i$ or $\boldsymbol{Y_i}$, the form of the pattern list is $\boldsymbol{L_{i2}} = [\boldsymbol{p_{i1}}, \ldots, \boldsymbol{p_{iM}}]$ containing individual success patterns.

Strategy Generator After completing persuasion tasks in Task-layer and accumulating enough success patterns, the Strategy Generator will summarize new strategies based on success patterns. Strategy Generator takes $\boldsymbol{L} = \bigcup_i \boldsymbol{L_{i2}}$ as input, and outputs $\boldsymbol{S^{new}} = [(\boldsymbol{s_i^{new}}, \boldsymbol{d_i}, \boldsymbol{L_i^s})]_{i=1}^{N_s}$, where $\boldsymbol{s_i^{new}}$ denotes a newly generated strategy, $\boldsymbol{d_i}$ denotes the persuasion domain to which the strategy belongs, $\boldsymbol{L_i^s}$ is the list of success patterns from which $\boldsymbol{s_i}$ is abstracted, and N_s is the number of new strategies. We prompt the LLM to identify shared persuasion principles or skills across different success patterns and abstract them into persuasion strategies. For each strategy, Strategy Generator assigns it to a domain based on its strategy description and success patterns.

We apply two constraints during strategy generation: (1) strategies within the same domain are required to be different to maintain diversity within each domain. (2) Similar strategies are allowed to appear in different domains to demonstrate cross-domain generalization.

Strategy Refiner Strategy Refiner updates the meta-strategy memory with new strategies and handles meta-strategy memory overflow. The **strategy updating** for each new strategy $\boldsymbol{s_i^{new}}$ in $\boldsymbol{S^{new}}$ can be represented as follows:

$$StrategyUpdater(\boldsymbol{s_i^{new}}, \boldsymbol{S^{meta}}[\boldsymbol{d_i}]) = \begin{cases} (\boldsymbol{s_i^{new}}, \boldsymbol{s_j^{now}}), & \exists \boldsymbol{s_j^{now}} \in \boldsymbol{S^{meta}}[\boldsymbol{d_i}] \mid same(\boldsymbol{S_i^{new}}, \boldsymbol{s_j^{now}}) \\ (\boldsymbol{s_i^{new}}, \boldsymbol{c_i}), & otherwise \end{cases} \quad (2)$$

where $\boldsymbol{d_i}$ denotes the domain of $\boldsymbol{s_i^{new}}$, $\boldsymbol{s_j^{now}}$ denotes a strategy in current strategy memory, $same(s_1, s_2)$ is a function that determines whether two strategies are identical, and c_i is the category of $\boldsymbol{s_i^{new}}$. (1) If there exists an identical strategy $\boldsymbol{s_j^{now}} \in \boldsymbol{S^{meta}}[\boldsymbol{d_i}]$, Strategy Updater outputs a pair $(\boldsymbol{s_i^{new}}, \boldsymbol{s_j^{now}})$. The success count and success patterns of $\boldsymbol{s_i^{new}}$ are then used to update the validity list and success pattern list of $\boldsymbol{s_j^{now}}$. Then, $\boldsymbol{s_i^{new}}$ is discarded. (2) Otherwise,

s_i^{new} is classified into the logical, emotional, or behavioral category and then inserted into the meta-strategy memory.

After the strategy updating, if the number of strategies in a domain exceeds a predefined upper bound U, Refiner will recruit an additional LLM to **handle overflow**. Only in this step will the meta-strategy memory be input into the LLM along with the success patterns of each strategy. The LLM will check if any similar strategies exist in the domain. Similar strategies are defined as those that: (1) employ similar methods to achieve similar goals; (2) could reasonably replace the other in its success patterns without changing the effect. The LLM outputs several groups of similar strategies. For each group, we retain the strategy with the highest validity among those satisfying $n_d > N_d$, ensuring that the high validity of the retained strategy is fully validated. If no strategy satisfies $n_d > N_d$, we retain the strategy with the largest n_d. If the number of strategies in the domain still exceeds the upper bound after deduplication, we further remove strategies with lower validity until the total number falls below U.

Clustering-based Hierarchization

After completing the iterative meta-strategy learning, we can observe that strategies from different domains may exhibit correlations and similarities, and strategies could also be macro or fine-grained. As a result, the meta-strategy memory still contains latent hierarchical structures that can be further extracted. We leverage clustering over persuasion domains and strategies to extract these hierarchical structures.

Domain Clustering For each domain, we concatenate the domain name with its strategies into a string and obtain its embedding vector. We then apply the K-means algorithm to cluster the vectors into several groups. We prompt the LLM to summarize a macro domain name for each group.

Strategy Clustering For the strategies within each domain, we also obtain their sentence embedding vectors and apply K-means clustering to form clusters of similar strategies. For strategies in each cluster, we prompt the LLM to determine whether they can be summarized by a macro-level strategy. Macro-level strategy is defined as an abstract strategy that captures the shared methods and goals of all the strategies in the group. If so, the LLM will generate the corresponding macro-level strategy; otherwise, the cluster is dissolved.

4 Experiment

Baselines

We compare our method with 5 baselines that are suitable for interactive persuasive dialogue: (1) **LLM+ICL**(In-Context Learning). We guide LLMs with high-quality dialogue examples from the DailyPersuasion dataset. (2) **LLM+ISR**(Intent-to-Strategy Reasoning) . We guide LLMs to analyze the persuadee's intent and select appropriate strategies before generating each response. 3) **DebateQD**. DebateQD [Reedi *et al.*, 2025] is originally designed for strategy extraction in the debate task. We adapt it to the DailyPersuasion dataset to extract persuasion strategies, which are then used to guide LLMs to perform persuasion tasks. 4) **PersuGPT**. PersuGPT[Jin *et al.*, 2024] is a persuasion model fine-tuned on the DailyPersuasion dataset based on the pre-trained LLaMA-2 Chat model.

We conducted experiments on the following base models: GPT-4o (mini), GPT-4.1 (nano), GPT-5 (nano), Gemini-2.0 (flash), and GPT-3.5 (turbo). Note that PersuGPT is fine-tuned on LLaMA-2 (Chat), which cannot serve as a base model for our method due to its limited context length. Additionally, the learning procedure of DebateQD requires access to token-level log probabilities from LLM outputs, which are not available through the Gemini-2-series APIs.

Evaluation

We apply four evaluation metrics to evaluate the persuasive dialogues. These metrics are evaluated by GPT-4o-mini and Deepseek-V3 to reduce potential model-specific biases. First, we design an evaluation metric to evaluate the Win Rate of persuasive dialogues: 1) **Win Rate**. Each dialogue is assigned a binary score by the LLM, where 1 denotes success and 0 denotes failure. The Win Rate is the percentage of persuasive dialogues with a score of 1. In addition, we adopt three evaluation metrics used in previous work [Zhang and Zhou, 2025]: 2) **Persuasiveness**. This metric assesses whether the persuader changes the persuadee's mind.3) **Empathy**. This metric assesses whether the persuader empathizes with the emotions of the persuadee. 5)**Helpfulness**. This metric assesses whether the persuader solves the user's problem. For clarity, we denote Win Rate as WR, Persuasiveness as Persu., Empathy as Empa. and Helpfulness as Help.

Implementation Details

We sample from the DailyPersuasion dataset[Jin *et al.*, 2024] to construct our learning and test sets. We first categorize the persuasion tasks by domain, and tasks involving multiple domains are randomly assigned to one. For strategy learning, we sample 7600 persuasion tasks with fewer than 3 domains each, and input them into our framework by domain. We set the batch size $B = 10$, the maximum number of strategies per domain $U = 24$, the maximum number of success patterns per strategy $W = 3$, and $N_d = 15$. For clustering, we use BGE-large-en-v1.5 to obtain a vector representation. Based on the within-cluster error and manual verification, we finally obtained 21 macro domains. In K-means clustering for strategies, we set K to N - 4, where N is the number of strategies within a domain. For testing, we randomly select 12 tasks from each domain, yielding 420 tasks for the test set, ensuring balanced evaluation across domains.

4.1 Performance on Different Base Models

We assess the effectiveness of PersuHMM across different base models. The results in Table 1 show that PersuHMM outperforms all baselines on Win Rate except for Gemini-2.0-flash as evaluated by Deepseek-V3. Analysis of failure cases shows that Gemini-2.0-flash sometimes overemphasizes following predefined persuasion steps, ignoring dialogue naturalness and the persuadee's immediate concerns. PersuHMM also achieves the highest average (from Persu. to Help.) across all base models, except for GPT-4.1-nano, evaluated by GPT-4o-mini. Compared with PersuGPT, PersuHMM achieves better performance based on GPT-4o-mini and Gemini-2.0-flash, and performs comparably to PersuGPT based on GPT-4.1-nano.

		GPT-4o-mini Evaluator					Deepseek-V3 Evaluator				
Model	**Method**	**WR**	**Persu.**	**Empa.**	**Help.**	**Avg.**	**WR**	**Persu.**	**Empa.**	**Help.**	**Avg.**
Llama-2	PersuGPT	61.90%	80.38	85.62	89.03	85.01	44.28%	81.30	84.57	87.01	84.29
GPT-4o	LLM+ICL	59.76%	80.10	85.68	89.29	85.02	47.14%	82.57	85.67	87.82	85.35
	LLM+ISR	63.57%	80.31	86.01	89.10	85.14	50.24%	**82.76**	85.95	87.57	**85.43**
	DebateQD	59.76%	80.05	85.61	89.43	85.03	48.10%	81.14	84.68	**87.83**	84.55
	PersuHMM	**81.19%**	**81.21**	**87.30**	**89.77**	**86.09**	**61.19%**	81.26	**87.20**	**87.83**	**85.43**
GPT-4.1	LLM+ICL	60.00%	80.10	84.61	**88.05**	**84.25**	44.05%	81.28	82.15	**87.06**	83.50
	LLM+ISR	59.05%	**80.31**	84.25	87.74	84.10	44.05%	**81.66**	79.19	86.75	82.53
	DebateQD	44.52%	77.12	79.63	86.62	81.12	38.10%	78.86	66.08	84.43	76.46
	PersuHMM	**61.67%**	79.48	**85.17**	87.70	84.12	**44.28%**	81.05	**83.43**	**87.06**	**83.85**
Gemini-2.0	LLM+ICL	62.86%	79.98	83.38	**89.09**	84.15	45.95%	81.47	79.86	86.94	82.61
	LLM+ISR	75.71%	81.63	84.87	88.20	84.90	**60.00%**	81.31	83.67	87.26	84.08
	PersuHMM	**76.19%**	**81.93**	**84.96**	88.45	**85.11**	55.24%	**81.73**	**84.45**	**87.77**	**84.65**

Table 1: Generalization across base models. Avg. denotes the average of the metrics from Persu. to Help.

			GPT-4o-mini Evaluator					Deepseek-V3 Evaluator				
Learner	**Persuader**	**Method**	**WR**	**Persu.**	**Empa.**	**Help.**	**Avg.**	**WR**	**Persu.**	**Empa.**	**Help.**	**Avg.**
GPT-4o	GPT-4.1	LLM+ICL	60.00%	80.10	84.61	**88.05**	84.25	44.05%	81.28	82.15	87.06	83.50
		LLM+ISR	59.05%	**80.31**	84.25	87.74	84.10	44.05%	**81.66**	79.19	86.75	82.53
		DebateQD	39.76%	77.75	77.24	85.20	80.06	33.57%	77.75	63.55	83.33	74.88
		PersuHMM	**61.19%**	80.05	**85.00**	87.75	**84.27**	**45.48%**	80.55	**83.11**	**87.69**	**83.78**
	GPT-5	LLM+ICL	77.38%	81.95	85.50	92.67	86.71	67.22%	82.79	85.24	89.22	85.75
		LLM+ISR	73.81%	82.01	85.89	92.38	86.76	68.10%	83.23	85.55	89.78	86.19
		DebateQD	74.76%	82.38	86.19	92.79	87.12	66.90%	83.88	84.49	**89.94**	86.10
		PersuHMM	**82.71%**	**87.71**	**87.02**	**92.82**	**89.18**	**72.38%**	**84.18**	**87.02**	89.42	**86.87**
	Gemini-2.0	LLM+ICL	62.86%	79.98	83.38	87.09	83.48	45.95%	81.47	79.86	86.49	82.61
		LLM+ISR	75.71%	81.63	84.87	**88.20**	**84.90**	60.00%	81.31	83.67	87.26	84.08
		DebateQD	64.29%	80.23	83.37	85.20	82.93	55.95%	81.67	82.88	87.58	84.04
		PersuHMM	**77.38%**	**81.65**	**84.98**	87.60	84.74	**60.48%**	**81.71**	**84.14**	**87.90**	**84.58**
	GPT-3.5	LLM+ICL	67.30%	80.38	85.03	89.28	84.90	44.52%	81.47	78.54	86.27	82.09
		LLM+ISR	69.21%	**80.69**	85.27	88.71	84.89	40.24%	81.06	73.09	**87.29**	80.48
		DebateQD	57.04%	80.31	84.75	**89.60**	84.89	37.62%	81.53	70.06	87.06	79.55
		PersuHMM	**71.12%**	80.45	**87.02**	88.97	**85.48**	**45.95%**	**82.63**	**84.54**	86.74	**84.64**

Table 2: Generalization across models. Avg. denotes the average of the metrics from Persu. to Help.

4.2 Strategy Generalization Across Models

We evaluate the generalization of our meta-strategy memory across different persuader models. We use the meta-strategy memory learned on GPT-4o-mini to guide different LLMs, with the results reported in Table 2. The results show that PersuHMM achieved the highest Win Rate across all persuader models, and also achieved the highest average (from Persu. to Help.) across nearly all persuader models, except for Gemini-2.0-flash, evaluated by GPT-4o-mini.

Notably, compared with Table 1, using the meta-strategy memory learned on GPT-4o-mini improves PersuHMM's performance across multiple metrics on GPT-4.1-nano and Gemini-2.0-flash. It suggests that, instead of learning a new meta-strategy memory for each model, choosing one appropriate model for learning enables more efficient reuse of meta-strategy memory. Furthermore, the results indicate that persuader models with stronger inherent persuasion and reasoning abilities (as measured by LLM+ISR) benefit more from our meta-strategy memory.

4.3 Generalization across Domains

We evaluate the Generalization of our method across persuasion domains with Deepseek-V3 as the evaluator. We randomly select 4 domains, and sample 400 single-domain tasks from each domain to prevent the model from being exposed to any test domain information while learning. With these datasets, we obtain 4 meta-strategy memories with PersuHMM and DebateQD, respectively. Then we test our method and the baselines on the Media, Science, Safety, and Charity domains. The results are reported in Table 3. Our method outperforms the baselines in nearly all learning–persuasion domain pairs on Win Rate, Persuasiveness, and Empathy, and performs better in half of the pairs on Helpfulness. Overall, the average of our method's metrics (from Persu. to Help.) achieves the best performance in nearly all domain pairs.

We also observe that the cross-domain generalizability of our method depends on the properties of the learning domains. For example, models learned on the Psychology and Business domain generally perform well across domains.

4.4 Ablation Study

We conducted an ablation study with Deepseek-V3 as the evaluator, as shown in Table 4. ⓪ is a baseline without any persuasion skills. ① discards the meta-strategy memory and prompts the Step Generator to generate persuasion steps and stage goals solely based on the task description. ② incorporates the meta-strategy memory learning process, but strategies are summarized by directly comparing our dialogues with ground-truth dialogues. ③ has the same components as the full model without the hierarchical structure of the meta-strategy memory. ④ is our full model.

Learning domain	Persuasion domain	Method (PersuHMM-)	ΔWR	Evaluation Metrics ΔPersu.	ΔEmpa.	ΔHelp.	ΔAvg.
Education	Media	LLM+ISR	+22.65%	+1.41	+1.95	+0.22	+1.19
		DebateQD	+12.27%	+0.08	+1.57	-0.11	+0.51
	Science	LLM+ISR	-1.89%	+0.39	+2.37	+0.44	+1.07
		DebateQD	+2.82%	-0.80	+2.87	-0.43	+0.55
	Safety	LLM+ISR	+5.66%	+0.23	+2.10	-0.27	+0.69
		DebateQD	+8.49%	+0.25	+1.86	+0.08	+0.73
	Charity	LLM+ISR	+7.54%	+0.81	+1.10	-0.13	+0.59
		DebateQD	+6.60%	-0.16	+1.52	-0.16	+0.40
Psychology	Media	LLM+ISR	+17.93%	+1.08	+2.60	-0.10	+1.19
		DebateQD	+7.55%	+0.18	+1.80	+0.09	+0.69
	Science	LLM+ISR	+7.55%	+1.30	+3.17	+0.22	+1.56
		DebateQD	+13.21%	+0.31	+2.93	-0.23	+1.00
	Safety	LLM+ISR	+3.78%	+0.19	+2.93	+0.07	+1.06
		DebateQD	+10.38%	-0.32	+2.50	+0.13	+0.77
	Charity	LLM+ISR	+4.71%	+1.32	+1.89	+0.09	+1.10
		DebateQD	+4.71%	+0.54	+2.30	-0.36	+0.83
Technology	Media	LLM+ISR	+24.53%	+1.36	+2.19	-0.05	+1.17
		DebateQD	+16.98%	+0.46	+1.43	-0.27	+0.54
	Science	LLM+ISR	-4.71%	+0.09	+2.12	-0.19	+0.67
		DebateQD	+6.61%	-0.63	+1.43	-0.33	+0.16
	Safety	LLM+ISR	+14.15%	+0.66	+2.52	-0.01	+1.06
		DebateQD	+11.32%	+0.16	+1.71	+0.15	+0.67
	Charity	LLM+ISR	+21.69%	+1.44	+0.40	+0.03	+0.62
		DebateQD	+13.21%	-0.85	+0.08	-0.25	-0.34
Business	Media	LLM+ISR	+12.27%	+0.85	+2.02	+0.10	+0.99
		DebateQD	+9.43%	+0.38	+2.08	-0.42	+0.68
	Science	LLM+ISR	+2.83%	+1.31	+2.02	+0.26	+1.20
		DebateQD	+9.43%	+0.45	+2.41	+0.44	+1.10
	Safety	LLM+ISR	+17.93%	+0.42	+2.05	+0.04	+0.84
		DebateQD	+21.70%	-0.16	+1.91	-0.33	+0.47
	Charity	LLM+ISR	+11.32%	+1.34	+1.12	+0.33	+0.93
		DebateQD	+8.49%	-0.06	+1.12	+0.01	+0.36

Table 3: Generalization across domains is represented by performance differences (Ours – Baseline), with green/red indicating superiority/inferiority, respectively. "Avg." averages all metrics from ΔPersu. to ΔHelp. LLM+ISR yields identical results within the same persuasion domain due to its learning-free nature.

	Module	WR	Pers.	Empa.	Help.	Token(L)	Token(T)
⓪	-	23.10%	80.29	82.40	86.35	-	2.92k
①	1	51.19%	81.03	86.58	87.25	-	6.35k
②	1, 2	58.33%	80.91	87.17	87.51	14.96k	10.58k
③	1, 2, 3	59.28%	81.02	87.14	87.77	16.54k	10.01k
④	1, 2, 3, 4	**61.19%**	**81.26**	**87.20**	**87.83**	17.15k	12.48k

Table 4: Ablation study. Module 1: Step generator. Module 2: Non-hierarchical meta-strategy memory. Module 3: Evaluator & Analyst. Module 4: Hierarchical meta-strategy memory. Token(L) and Token(T) denote average tokens per sample during learning and testing. No learning for ⓪ and ①.

Comparing ① and ② shows that the meta-strategy memory substantially improves Win Rate and Empathy. Adding the Evaluator and Analyst (② vs. ③) slightly improves Win Rate, Persuasiveness, and Helpfulness. Introducing the hierarchical structure (③ vs. ④) further improves Win Rate and Persuasiveness at the cost of a slight reduction in Coherence. Overall, all modules contribute positively to win rate and are effective in achieving higher task performance.

Regarding token cost (Table 4), during learning, ③ consumed 10.56% more tokens than ② (+1.63% win rate), while ④ achieved a further 3.22% win rate gain with only 3.69% more tokens, demonstrating the cost-effectiveness of the hierarchical structure. During testing, ⓪–② showed proportional performance–cost trade-offs. Notably, ③ reduced token consumption while improving win rate, likely due to fewer dialogue rounds. ④ incurred 24.8% higher token cost than ③ with a 3.22% win rate improvement. Generally, given the difficulty of improving upon already high win rates, the additional token costs are justified.

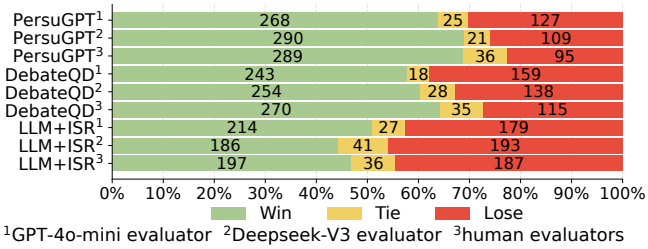

Figure 4: **LLM and human** preference results. Weighted Cohen's κ: Human vs. Deepseek-V3 = 0.644, Human vs. GPT-4o-mini = 0.510, both indicating agreement.

4.5 LLM and Human Preference Study

We conduct a preference study with LLM and human evaluators, using Deepseek-V3 and GPT-4o-mini as LLM evaluators and recruiting three human volunteers. Prompts follow prior work [Anonymous, 2026]. For each test instance, evaluators judge two anonymous dialogues generated by different methods as win, tie, or lose. Three method pairs are compared: PersuGPT vs. PersuHMM (GPT-4o-mini), DebateQD vs. PersuHMM (both GPT-4o-mini), and LLM+ISR vs. PersuHMM (both Gemini-2.0-flash). As shown in Figure 4, PersuHMM is preferred over nearly all baselines under both LLM and human evaluation, indicating higher-quality persuasive responses in practice.

To quantify LLM–human consistency, we compute the weighted Cohen's kappa k_w [Cohen, 1968], treating the three labels as ordinal categories (lose $<$ tie $<$ win) with quadratic weighting. The k_w between human evaluators and DeepSeek-V3 is 0.644 (high agreement), and 0.510 with GPT-4o-mini (moderate agreement). Overall, the results demonstrate agreement between human and LLM evaluators. In particular, DeepSeek-V3's preferences exhibit higher agreement with human judgments, making it a more reliable evaluator.

5 Conclusion

We present PersuHMM, a meta-learning framework that learns and organizes hierarchical persuasion strategies through iterative task execution and meta-refinement. The autonomously built meta-strategy memory proves effective in boosting persuasive performance and generalizability, both across domains and across models.

Acknowledgements

We would like to thank anonymous reviewers for their valuable comments. The authors acknowledge financial support from the National Natural Science Foundation of China (62176053).

Contribution Statement

Yanyue Zhang and Zihao Wang contributed equally to this work.

References

[Altay *et al.*, 2023] Sacha Altay, Anne-Sophie Hacquin, Coralie Chevallier, and Hugo Mercier. Information delivered by a chatbot has a positive impact on covid-19 vaccines attitudes and intentions. *Journal of Experimental Psychology: Applied*, 29(1):52, 2023.

[Anonymous, 2026] Anonymous. MA^{2}p: A metacognitive autonomous intelligent agents framework for complex persuasion. In *Submitted to ACL Rolling Review - January 2026*, 2026. under review.

[Bai *et al.*, 2025] Hui Bai, Jan G Voelkel, Shane Muldowney, Johannes C Eichstaedt, and Robb Willer. Llm-generated messages can persuade humans on policy issues. *Nature Communications*, 16(1):6037, 2025.

[Cohen, 1968] Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.

[Costello *et al.*, 2024] Thomas H Costello, Gordon Pennycook, and David G Rand. Durably reducing conspiracy beliefs through dialogues with ai. *Science*, 385(6714):eadq1814, 2024.

[Dönmez and Falenska, 2025] Esra Dönmez and Agnieszka Falenska. "i understand your perspective": Llm persuasion through the lens of communicative action theory. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15312–15327, 2025.

[Hackenburg *et al.*, 2024] Kobi Hackenburg, Ben M Tappin, Paul Röttger, Scott Hale, Jonathan Bright, and Helen Margetts. Evidence of a log scaling law for political persuasion with large language models. *arXiv preprint arXiv:2406.14508*, 2024.

[Jin *et al.*, 2023] Chuhao Jin, Yutao Zhu, Lingzhen Kong, Shijie Li, Xiao Zhang, Ruihua Song, Xu Chen, Huan Chen, Yuchong Sun, Yu Chen, et al. Joint semantic and strategy matching for persuasive dialogue. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4187–4197, 2023.

[Jin *et al.*, 2024] Chuhao Jin, Kening Ren, Lingzhen Kong, Xiting Wang, Ruihua Song, and Huan Chen. Persuading across diverse domains: a dataset and persuasion large language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1678–1706, 2024.

[Mishra *et al.*, 2022] Kshitij Mishra, Azlaan Mustafa Samad, Palak Totala, and Asif Ekbal. PEPDS: A polite and empathetic persuasive dialogue system for charity donation. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 424–440, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[Potter *et al.*, 2024] Yujin Potter, Shiyang Lai, Junsol Kim, James Evans, and Dawn Song. Hidden persuaders: Llms' political leaning and their influence on voters. *arXiv preprint arXiv:2410.24190*, 2024.

[Qin *et al.*, 2024] Peixin Qin, Chen Huang, Yang Deng, Wenqiang Lei, and Tat-Seng Chua. Beyond persuasion: Towards conversational recommender system with credible explanations. *arXiv preprint arXiv:2409.14399*, 2024.

[Reedi *et al.*, 2025] Aksel Joonas Reedi, Corentin Léger, Julien Pourcel, Loris Gaven, Perrine Charriau, and Guillaume Pourcel. Optimizing for persuasion improves llm generalization: Evidence from quality-diversity evolution of debate strategies. *arXiv preprint arXiv:2510.05909*, 2025.

[Rogiers *et al.*, 2024] Alexander Rogiers, Sander Noels, Maarten Buyl, and Tijl De Bie. Persuasion with large language models: a survey. *arXiv preprint arXiv:2411.06837*, 2024.

[Saenger *et al.*, 2024] Till Raphael Saenger, Musashi Hinck, Justin Grimmer, and Brandon M. Stewart. AutoPersuade: A framework for evaluating and explaining persuasive arguments. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16325–16342, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[Sakurai and Miyao, 2024] Hiromasa Sakurai and Yusuke Miyao. Evaluating intention detection capability of large language models in persuasive dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1635–1657, 2024.

[Salvi *et al.*, 2024] Francesco Salvi, Manoel Horta Ribeiro, Riccardo Gallotti, and Robert West. On the conversational persuasiveness of large language models: A randomized controlled trial. *arXiv preprint arXiv:2403.14380*, 2024.

[Samad *et al.*, 2022] Azlaan Mustafa Samad, Kshitij Mishra, Mauajama Firdaus, and Asif Ekbal. Empathetic persuasion: Reinforcing empathy and persuasiveness in dialogue systems. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 844–856, Seattle, United States, July 2022. Association for Computational Linguistics.

[Stiff and Mongeau, 2016] James B Stiff and Paul A Mongeau. *Persuasive communication*. Guilford Publications, 2016.

[Wan *et al.*, 2025] Chenwei Wan, Matthieu Labeau, and Chloé Clavel. Emodynamix: Emotional support dialogue strategy prediction by modelling mixed emotions and discourse dynamics. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1678–1695, 2025.

[Wu *et al.*, 2025] Shenghan Wu, Yimo Zhu, Wynne Hsu, Mong-Li Lee, and Yang Deng. From personas to talks: Revisiting the impact of personas on llm-synthesized emotional support conversations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5439–5453, 2025.

[Xie *et al.*, 2025] Henry J Xie, Jinghan Zhang, Xinhao Zhang, and Kunpeng Liu. Distilling empathy from large language models. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 343–354, 2025.

[Zhang and Zhou, 2025] Dingyi Zhang and Deyu Zhou. Persuasion should be double-blind: A multi-domain dialogue dataset with faithfulness based on causal theory of mind. *arXiv preprint arXiv:2502.21297*, 2025.

[Zhang *et al.*, 2025] Yazhou Zhang, Mengyao Wang, Youxi Wu, Prayag Tiwari, Qiuchi Li, Benyou Wang, and Jing Qin. Dialoguellm: Context and emotion knowledge-tuned large language models for emotion recognition in conversations. *Neural Networks*, page 107901, 2025.