Universiteit Leiden

Master Computer Science

Improving Empathetic Dialogue through Prompt Augmentation and Vector Modulation

Name:	Hanlei Zhu
Student ID:	s3975789
Date:	27/08/2025
Specialisation:	Data Science: Computer Science
1st supervisor:	Zhaochun Ren
2nd supervisor:	Flor Miriam Plaza del Arco

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Empathetic dialogue generation is a challenging task in conversational AI. Large Language Models (LLMs) perform very well at fluency, but they struggle to capture specific user emotions and instead produce generic responses. We fix this gap with a hybrid approach that enhances empathetic responses in an In-Context Learning (ICL) framework. We investigate two strategies: (1) augmenting input prompts with external knowledge and (2) directly steering the model's internal reasoning process. For the input augmentation strategy, our experiments compare narrative knowledge (from COMET) with lexical features (from VAD, ConceptNet). For the steering internal reasoning strategy, we introduce Vector-Modulated Collaborative Prompting (VMCP), a two-stage method for internal emotional guidance. Our key finding is that combining external narrative context with internal vector steering creates a powerful combination effect, achieving the best accuracy of 45.96% on the EmpatheticDialogues dataset. This hybrid model outperforms either approach separately. This means that providing both rich context and precise internal control is essential for generating empathetic dialogue.

Contents

Abstract **2**

1 Introduction **5**

2 Background and Related Work **6**

2.1 Empathetic Dialogue Generation 6

2.2 In-Context Learning for Empathetic Dialogue 6

2.3 Knowledge-Augmented Empathetic Dialogue 7

2.3.1 Narrative and Commonsense Knowledge 7

2.3.2 Lexical and Semantic Knowledge 7

2.4 Controllable Generation and Model Steering 7

2.5 Research Gaps and Our Contributions 8

3 Methods **9**

3.1 Baseline Framework: ICL 9

3.2 Prompt-based Augmentation Strategies 9

3.2.1 Prompt Enrichment with COMET 10

3.2.2 Prompt Enrichment with Lexical Features 11

3.3 Vector-Modulated Collaborative Prompting (VMCP) 13

3.3.1 Formulating Guidance Vectors from Retrieved Examples 13

3.3.2 Two-Stage Injection of VMCP: EGI and AIR 14

4 Experiments and Results **16**

4.1 Experimental Setup 16

4.2 Baseline Performance 17

4.3 Main Results 18

4.3.1 Overall Performance and Method Comparison 18
4.3.2 Cross-Model Generalization of Core Components 18
4.3.3 Performance of the Combined Model 19
4.4 Analysis of Design Choices 21
4.4.1 Evaluation of Prompt-based Augmentation Strategies 21
4.4.2 Details of Design Choices for External Knowledge 22
4.4.3 Ablation Study of VMCP Components 22
4.5 Qualitative Analysis 23

5 Discussion **27**

5.1 Explanation of Prompt-based Augmentation Strategies 27
5.1.1 The 'Role Mismatch' Challenge in COMET Knowledge 27
5.1.2 Comparative Analysis of Narrative Knowledge vs. Lexical Features 28
5.1.3 Analysis of VAD and ConceptNet 28
5.2 Core Mechanism and Deeper Impact of VMCP 30
5.2.1 VMCP's Performance Compared to Prompting 30
5.2.2 Combining External Prompts and Internal Control 30
5.2.3 Analysis of Accuracy vs. Diversity 31
5.2.4 The Critical Role of Adaptive Internal Refinement (AIR) 31
5.3 Generalization to Cross-Domain Data 32
5.4 Limitations and Future Work 32

6 Conclusion **33**

A Supplementary Experimental Details **36**

A.1 Detailed Prompt Augmentation Results on All Models 36
A.2 VAD Keyword Selection 36
A.3 COMET Knowledge Filtering 38

Chapter 1 Introduction

Conversational AI is becoming more advanced, with Large Language Models (LLMs) being able to generate fluent and clear text [1]. LLMs can interact with users on an emotional level, and we call this empathetic dialogue. Empathetic dialogue systems try to understand a user's emotions and respond with empathy. This is important for mental health, customer service, and human-computer interaction [2] [3].

However, creating AI that generates truly empathetic responses is still a challenge. We can input prompts into LLMs to perform tasks using In-Context Learning (ICL) [1], but their responses often lack emotional depth [4]. They might use generic phrases like "I'm sorry to hear that", but miss the specific and detailed emotions of the users. So there is a clear difference between a response with only the correct grammar and one that shows real understanding.

There are two main strategies to solve this problem: augmenting the model's prompt input [5] [6] and controlling its internal reasoning processes [7]. The first strategy adds external knowledge to the prompt to give the model more context. However, it is unclear which type of knowledge works best, whether it is narrative, commonsense, or lexical facts. The second strategy is to control the model's generation process directly. The main challenge is how to guide the model's emotion without breaking its reasoning.

Our work introduces a new hybrid approach to solve these problems. We combine a new internal control method with prompt augmentation strategies. Our experiments show that this combined method, which provides the model with both narrative context and direct emotional guidance, substantially improves empathetic response generation.

The rest of this paper is structured as follows. Chapter 2 reviews past work. Chapter 3 explains our methods. Chapter 4 shows our experiments and results. Chapter 5 discusses what our results mean, and Chapter 6 concludes the paper.

Chapter 2 Background and Related Work

This chapter reviews the development of empathetic dialogue. It talks about the history from rule-based systems to modern neural models. Then it focuses on the In-Context Learning that we use as our baseline in the study. Building on the baseline, we bring out the main strategies we use for our work: Knowledge Augmentation and Vector Modulation. In the end, we highlight the weaknesses in current research and present our main contribution.

2.1 Empathetic Dialogue Generation

The main challenge in this field is to avoid generic responses like "I'm sorry to hear that" and instead generate specific replies. Researchers first tried to solve this with hand-crafted rules. More recently, work has changed to using end-to-end neural network models.

Early empathetic dialogue systems mainly relied on hand-crafted rules [8]. With the development of deep learning, researchers changed to using end-to-end neural network models. At this time, much work focused on improving the system's empathetic capabilities through more complex model architectures. MoEL [9] proposes to train specialized decoders ("listeners") for different emotions. MIME [10] introduces the concept of "emotion mimicry" and adjusts the degree of mimicry based on the positive or negative nature of the emotion. EmpDG [11] uses multi-granularity emotional factors and adversarial learning to capture more subtle emotional changes. These works promoted the field of empathetic response generation, but they mainly focused on the fine-tuning paradigm for specific models.

2.2 In-Context Learning for Empathetic Dialogue

In recent years, Large Language Models (LLMs) bring a new approach to AI, which is In-Context Learning (ICL). ICL does not require model parameter updates. Instead, it guides a model on a specific task by giving a few examples in the prompt [1].

ICL is applied to empathetic response generation by retrieving examples that are semantically similar to the user's query [4]. The model then follows the pattern of these examples to generate a response. However, relying on semantic similarity can be a problem, as semantically similar examples can be emotionally different [12]. Our work builds upon the ICL method but starts from a baseline without examples to test other enhancement methods.

2.3 Knowledge-Augmented Empathetic Dialogue

To help models understand the deeper meaning of a dialogue better, many researchers have injected external knowledge into empathetic dialogue systems. The two main types of external knowledge we use are commonsense knowledge and semantic knowledge.

2.3.1 Narrative and Commonsense Knowledge

Many researchers add commonsense knowledge to their models to improve empathetic responses. They often use large knowledge bases like ATOMIC [13] and COMET [14] for this. For example, the CEM method [5] uses commonsense from COMET to help the model better understand the user's situation. Other works also use this idea to make sure the model's response fits the user's situation and feelings [8, 15]. These examples show that providing commonsense knowledge is an effective way to improve empathy in dialogue systems.

2.3.2 Lexical and Semantic Knowledge

Other methods focus on word-level features and their meanings. KEMP [6] is a representative work. It improves the model in two ways. First, it uses the VAD emotion lexicon [16] to find core emotional keywords. Second, it uses the ConceptNet [17] knowledge graph to expand the keywords' semantic associations and build a context graph to guide generation. ESCM [18] also examines grammatical correlations between emotional words and semantic roles. These methods improve empathy by helping the model understand key emotional words.

2.4 Controllable Generation and Model Steering

Besides using knowledge to optimize the prompt, another way is to control the model's generation process directly. This is known as "controllable generation". While traditional methods do this through prompt designing, as the knowledge augmentation method described in Chapter 2.3, more recent work tries to steer the model's internal state directly.

For example, some methods are inspired by psychology. EmpSOA [19] controls the model's understanding of "self" and "other" to guide empathy. IAMM [7] is based on human memory to capture key emotional information. Other methods use different architectures. For instance, DIFFUSEMP [20] uses a diffusion model and fine-grained control signals to manage the response style.

These studies show that directly steering the model's internal state is a promising research direction. Our VMCP method follows this path by injecting a guidance vector directly into the model's hidden layers.

2.5 Research Gaps and Our Contributions

Reviewing the related work, we found these research questions. First, what kind of external knowledge is the most helpful for ICL? Second, how to steer a model's internal feelings? And finally, what happens if we combine these two approaches?

Our paper answers these questions.

1. We tried different prompt augmentation strategies, and the result is that providing narrative context (from COMET-Situation) works much better than using lexical features.

2. We proved that our internal vector injection method (VMCP) is an effective way to guide the model's emotion from the inside.

3. Our key finding is that combining these two methods creates substantially better results than either method alone.

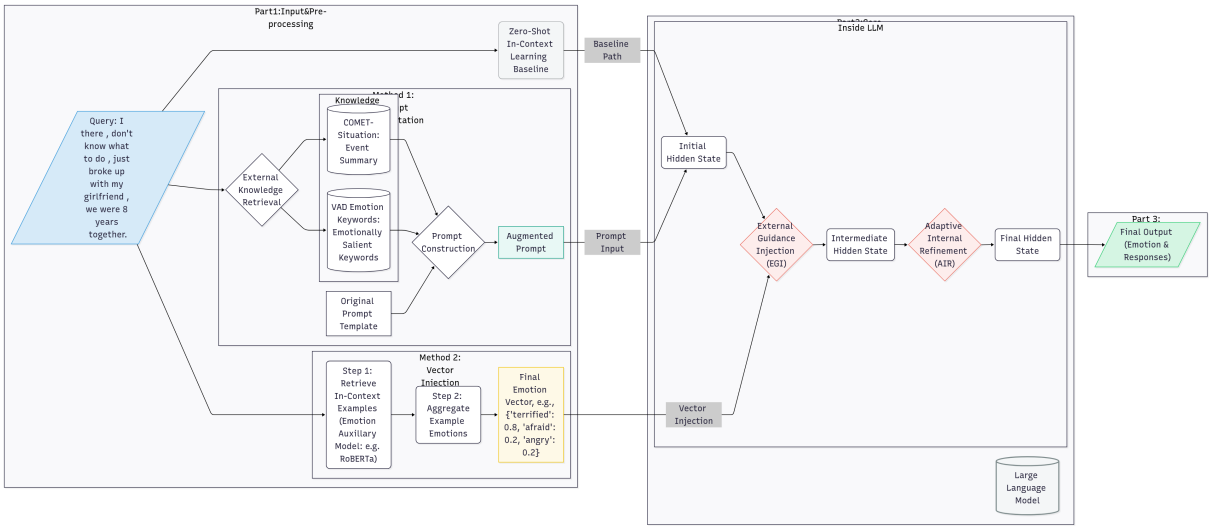

Figure 1: Architecture of our proposed prompt augmentation & VMCP framework.

Chapter 3 Methods

This chapter describes the methods we use to improve Empathetic Dialogue Generation. Our core idea is to help the model generate more empathetic responses. We do this by improving its ability to understand and recognize fine-grained emotions.

Our research has two main stages. First, we explore Prompt-based Retrieval Augmentation, enriching the model's input prompt with external knowledge. Second, we propose a method called Vector-Modulated Collaborative Prompting (VMCP), which uses vector injection and addition to control the emotions generated by reasoning. The overall process of our entire research method is shown in Figure 1.

3.1 Baseline Framework: ICL

We use a Zero-Shot In-Context Learning (ICL) framework as the baseline of our research. This is a specific type of ICL where the prompt provides the model with detailed instructions but no examples. This approach allows us to measure the model's capabilities based on instructions alone. Without examples, we can more accurately evaluate the impact of our proposed methods, independent of the complex effects of examples.

3.2 Prompt-based Augmentation Strategies

Building on the ICL framework, we first tested prompt augmentation by adding external knowledge to the Prompt. Our goal was to find the most effective source of knowledge for this method through experiments. We mainly used two sources of knowledge: common-

sense knowledge from COMET[5] and semantic knowledge from VAD and ConceptNet[6].

3.2.1 Prompt Enrichment with COMET

To help the model better understand the cause, intent, and emotion in a dialogue, we use the COMET commonsense knowledge base. In our study, we directly use the `dataset_preproc.p` file from the CEM project[5][1], which was pre-processed on the EmpatheticDialogues dataset. This file provides different types of commonsense knowledge from the COMET model for each dialogue, including *situation*, *xReact* (emotional reactions), and *xIntent* (intentions).

The core of our method is how to filter and insert the most relevant knowledge for each query from this raw knowledge base. We process in these three steps:

Step 1: Dialogue Matching

First, we use the all-MiniLM-L6-v2 model to calculate the similarity between the dialogue content in `dataset_preproc.p` and the dialogue content of each Query in our test set. If the cosine similarity is greater than 0.95, we determine that the current dialogue is the same one. This allows us to extract the COMET knowledge obtained from `dataset_preproc.p` and apply it to the prompt.

Step 2: Relevance Filtering by 'Situation'

While we get the COMET knowledge for each Query, there is a challenge in how to use this external knowledge smartly. Our task is to identify the emotion of the "last speaker"; however, the emotional knowledge that COMET gives is not from the "last speaker" but the event's "experiencer". So we need to identify the COMET knowledge whose event's "experiencer" is the "last speaker" of the query. To achieve this, we extract its 'situation text', which represents the situation of the event's "experiencer". We then calculate its semantic similarity with the full utterances of the last speaker in the current query. We only consider the COMET knowledge relevant if the cosine similarity is above a threshold of 0.5, which serves as proof that the event's "experiencer" is the "last speaker" in the dialogue context. Otherwise, the external knowledge of COMET for that query is left empty. This mechanism acts as a gatekeeper to ensure that all injected COMET knowledge is relevant.

Step 3: Preparing the Knowledge for the Prompt

For knowledge that passes the filter, we process it based on its type:

1. For situation knowledge: We perform pronoun conversion. We use a regular expression function to change first-person pronouns (like I, my, we, our) to a neutral third-person perspective (like the speaker, their). This avoids confusion with the dialogue's point of view. The processed text is formatted as "Event Summary: [text]".

[1]The CEM project and its pre-processed data are available at: `https://github.com/Sahandfer/CEM`

2. For xReact and xIntent knowledge: We apply the same relevance filter and also re-rank the phrases. We extract the list of xReact phrases. After cleaning the text (e.g., removing "none"), we calculate the similarity of each phrase with the last speaker's utterances. We then rank these phrases by their similarity score. This makes sure the most relevant phrase is ranked first. Finally, the sorted phrases from xReact and xIntent are presented as "Potential Reactions: [phrase1, phrase2, ...] and "Speaker's Inferred Intent: [phrase1, phrase2, ...]".

After preparing each type of knowledge, we place this processed knowledge as "External Knowledge" after the 'Dialogue context' in the Prompt, along with a guiding sentence, as shown in Chapter 4.1 ('Augmented Prompt Examples').

For our cross-domain tests on the EDOS dataset, the dialogue-matching method was not effective. So, we used a more direct retrieval method. Instead of matching whole dialogues, we took only the last speaker's utterance from an EDOS dialogue. We used this utterance as a query to search the entire COMET knowledge base for the most similar situation description. To do this, we computed embeddings for all situation texts and used a 0.5 cosine similarity threshold to define a match. This approach connects the speaker's final utterance directly to the best available background knowledge.

In these methods, we focus our work on the 'Situation' knowledge. We believe this approach is more effective because providing a complete story as context is suitable for the model's reasoning process. For comparison, we also prepared the lexical xReact and xIntent knowledge. We detail the performance of these different augmentation methods in Chapter 4.

3.2.2 Prompt Enrichment with Lexical Features

Unlike the narrative context provided by the COMET situation, we also tested a method that uses VAD and ConceptNet as external knowledge, which is based solely on lexical features. This strategy was inspired by the KEMP method [6]. In our implementation[2], we directly used the data resources from the KEMP project on the EmpatheticDialogues dataset. Our work is split into two Stages:

Stage 1: Knowledge Extraction with VAD Keyword

The goal of this stage is to select five keywords with high emotional intensity from the full utterances of the last speaker. The process is as follows.

1. Selection based on Emotion Intensity: We first use the `VAD.json` file from the KEMP project. This file contains emotion scores (Valence, Arousal, Dominance) for words in the EmpatheticDialogues dataset. We use the formula from KEMP, shown in

[2]The KEMP project and pre-processed data are available at: `https://github.com/qtli/KEMP`

Equation (1), to define the emotion intensity score $\eta(w)$ for each word:

$$\eta(w) = \text{min-max}(\left\| V_a(w) - \frac{1}{2}, \frac{A_r(w)}{2} \right\|_2) \quad (1)$$

Using this score, we select the top 10 words with the highest emotion intensity as candidates for each dialogue context.

2. Re-ranking based on Contextual Relevance: To make sure the keywords are relevant to the dialogue, we use a *RoBERTa-based model (stsb-roberta-base)* to calculate the semantic similarity between each candidate word and the dialogue context. We then re-rank the candidates based on this score.

3. Removing Similar Keywords: To ensure keyword diversity, we filter the candidate list by removing any word with a cosine similarity above 0.85 to an already selected keyword. Finally, we choose the top 5 keywords as the final output, and format them as "Emotionally salient keywords: [word1, word2, ...]". We put these words as "External Knowledge" after the "Dialogue context" in the Prompt, as shown in Chapter 4.1 ('Augmented Prompt Examples').

Stage 2: Knowledge Expansion with ConceptNet

After getting high-quality VAD keywords, we use the ConceptNet knowledge graph to expand them. We use the pre-processed *ConceptNet_VAD_dict.json* file from the KEMP project[3]. This file contains relevant ConceptNet triples for each emotional word. The process is as follows:

1. Relation Retrieval and Filtering: Using the VAD keywords from Stage 1 as 'head concept', we retrieve all their related knowledge triple (head concept, relation, tail concept) from the *ConceptNet_VAD_dict.json* file. Then we filter them to get only the tuples with strong semantic relations (e.g., IsA, RelatedTo, HasProperty, Synonym, SimilarTo).

2. Natural Language Conversion and Relevance Ranking: We convert these triples into natural language phrases (e.g., "terrified is a synonym of frightened"). Then, we use a RoBERTa-based model to calculate the similarity of each phrase with the dialogue context and rank them.

3. Final Selection and Integration: For each chosen keyword, we select the most relevant top-k (k = 1) phrases. These selected phrases from ConceptNet are then injected into the Prompt. We format them as: "The following related concepts are inferred from ConceptNet to support emotion understanding:[concept1, concept2, ...]".

Our method uses the VAD and ConceptNet lexicons from the KEMP project [6], which were created using only the ED dataset. This creates a mismatch when we test on the EDOS dataset. There are two issues: (1) some words from an EDOS dialogue may not

[3]The KEMP project and pre-processed data are available at: `https://github.com/qtli/KEMP`

exist in our lexicons, and (2) even for words that do exist, their pre-computed VAD scores are based on the ED context and may not be suitable for the EDOS context.

Our method proceeds by accepting this limitation. For a given word from an EDOS dialogue, we look it up in our ED-based *VAD.json*. If the word is not found, we discard it. If it is found, we use its VAD scores to calculate the emotion intensity score, and then use that keyword to find relations in our ED-based *ConceptNet_VAD_dict.json*. We believe this is a good trade-off. We prioritize the quality of the keywords we inject, not the quantity.

Finally, these external sources of knowledge were integrated into the Prompt, as shown in Chapter 4.1 ('Augmented Prompt Examples'), with external knowledge from ConceptNet.

In Chapter 4, we evaluate these lexical strategies and compare them against the narrative approaches.

3.3 Vector-Modulated Collaborative Prompting (VMCP)

Vector-Modulated Collaborative Prompting (VMCP) is an internal method that directly changes the model's reasoning process. It is different from Prompt augmentation, which changes the input. The core idea of VMCP is to inject a guidance vector into the model's hidden layers. This vector is created dynamically and injected at a key moment to precisely control the model's emotion generation. The method has two main steps.

3.3.1 Formulating Guidance Vectors from Retrieved Examples

The goal of this step is to create a unique "composite guidance vector" (controller) for each test sample. This vector guides the model's behavior in the next stage. The process is as follows:

1. **Getting the Data:** We use the *ed_tst.json* dataset prepared by the EICL framework [12]. This file gives us two key things for each query we need to process: five "emotionally similar examples" and their corresponding "dynamic soft labels".
2. **Finding the Main Emotion from Examples:** We don't use the ground truth labels of the examples. Instead, we use their "dynamic soft labels" to figure out the main emotional signal. For each of the five examples, we find its most likely emotion based on its soft label distribution. Then, we use a voting system where each of the five examples "votes" for its main emotion. We sum these votes to get an overall emotional tendency for the current query (e.g., 'sad': 0.6, 'lonely': 0.2).
3. **Creating the Guidance Vector:** Finally, we use this emotional tendency to build the controller vector. We have a pre-generated library (*rep_dict.p*) that stores a standard vector for each emotion type. Based on our calculated tendency (e.g., 60%

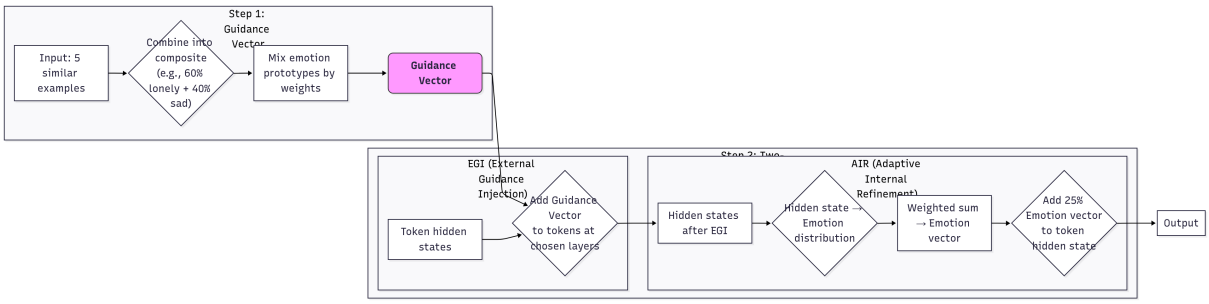

Figure 2: Architecture of our proposed prompt augmentation & VMCP framework.

sad, 20% lonely), we retrieve the standard vectors for "sad" and "lonely" and mix them using these weights. This creates a "composite guidance vector" for the current dialogue.

3.3.2 Two-Stage Injection of VMCP: EGI and AIR

We apply the controller vector using a two-stage mechanism: "External Guidance Injection" and "Adaptive Internal Refinement". The process is shown in Figure 2.

Stage 1: External Guidance Injection (EGI)

The goal of External Guidance Injection (EGI) is to give the model an initial emotional direction based on the dominant emotion of similar Examples. We use the controller vector as an instruction for direction and magnitude. We use a hyperparameter, multi_coeff, to scale its overall strength.

At specific steps in the model's generation process (e.g., when forward_id is 3 or 4), we inject this controller vector by adding it to the hidden state of the model's layers (i.e., all wrapped layers). This gives the model a clear initial emotional direction.

Stage 2: Adaptive Internal Refinement (AIR)

The Adaptive Internal Refinement (AIR) stage is a refinement step that follows immediately after EGI within the same generation step. Its purpose is to adjust the model's emotional direction based on its current state.

First, we check the model's current emotional state. We calculate the dot product between the current hidden state and all emotion prototype vectors. We then pass the similarity scores through a softmax function to create a normalized probability distribution. This distribution shows the model's current leaning towards each emotion.

Next, we use this distribution to create the 'Refinement Vector'. This vector is a weighted sum of the original emotion prototypes, using the probabilities as the weights.

Finally, we multiply this 'Refinement Vector' by coeff = 0.25. Then, we inject it back into the hidden state.

Through the guidance of EGI and the refinement of AIR, the VMCP method aims to achieve precise and reliable control over the model's emotion generation path.

Chapter 4 Experiments and Results

4.1 Experimental Setup

Datasets

We evaluate our methods on two benchmark datasets: EmpatheticDialogues (ED) [3] and EDOS [21]. Our core experiments use the ED dataset, which contains about 25,000 dialogues covering 32 emotions. We also use the EDOS dataset, with 10,000 dialogues across 42 emotions, to test how well our methods generalize.

This design for our EDOS tests has two goals. First, it shows how well our methods work on new data when the knowledge source is not a perfect fit. Second, it replicates a real-world situation where we apply an old knowledge base to a new problem because we don't have the resources to build a new one. This makes our test on EDOS more demanding.

We use three main data sources: 1) A pre-processed test set (*ed_tst.json*), originally prepared for the EICL framework [12], which contains dialogue query and corresponding retrieved examples; 2) CEM's COMET knowledge base (`dataset_preproc.p`), from which we extract commonsense knowledge like situation and xReact [5]; 3) KEMP's VAD/ConceptNet resources, using its `VAD.json` and *ConceptNet_VAD_dict.json* files [6].

Task Design and Evaluation Metrics

The output format for all tasks follows the structure: **Emotion: [a single inferred emotion] — Response: [a concise and appropriate response]**. This requires the model to do both emotion classification and response generation.

It is difficult to measure the quality of an empathetic response with a single score. So we use both Accuracy and Distinct-1/Distinct-2 to evaluate our model's performance [6].

1. **Accuracy.** We use Accuracy to check the first task (emotion prediction). This metric shows how often our model accurately predicts the correct emotion for a given 'Dialogue context'. A high Accuracy score means the model correctly understands the user's feelings. This is the first step to writing a good response.
2. **Distinct-1 and Distinct-2.** We use these two metrics to check the second task (response writing). They count the number of unique words (Distinct-1) and unique pairs of words (Distinct-2) in the generated responses. These scores show if the model's responses are diverse. High Distinct scores suggest the model generates more interesting content.

Together, these metrics give a clear picture of our model's performance. Accuracy tells us if the emotional understanding is correct. Distinct tells us if the language is creative.

Models and Implementation Details

We select the large language models **Mistral-Nemo**[4], **Phi-3.5-mini**[5], and **Llama3.1-8B**[6]. This is to check the performance of different methods across different model architectures. All experiments were run on **two NVIDIA GeForce RTX 3090 GPUs**, each with 24GB of memory. Common settings include a Batch Size of 2 on Mistral-Nemo, 16 on Phi-3.5-mini and Llama3.1-8B, and a Max New Tokens of 100.

For our hyperparameters, we adopted settings based on prior work and early tests. We set the COMET relevance threshold to 0.5. For the number of VAD keywords, we tested several values and selected top-k=5 as it provided a good balance of relevance and coverage. We explain the reasons for these choices in more detail in Appendix A. For our VMCP method, we set the injection layers from -3 to -17. Based on preliminary experiments and the established configuration for this method, we use an EGI coefficient (multi_coeff) of 1.5 and an AIR coefficient (coeff) of 0.25, as these values provide consistent performance.

Prompt Template Example

To give a more intuitive understanding, an example of a final, more complex prompt (ICL + COMET-situation + VAD) is shown below:

Augmented Prompt Example

```
[INST]
   Infer the emotion of the dialogue context...
   - Dialogue context: The conversation history...
   - Emotion labels: surprised, excited, annoyed, proud...
   - Note the 'Event Summary' if provided, as it gives narrative context.
   - Pay attention to the 'Emotionally salient keywords'...
   - Response Format: Emotion: [...] Response: [...].
    Dialogue context: yeah about 10 years ago i had a horrifying
experience...

   Event Summary: The speaker just broke up with...

   Emotionally salient keywords: years, together...
[/INST]
```

4.2 Baseline Performance

This section sets the performance "starting line" for all later experiments. We first evaluated the original Zero-Shot ICL framework on multiple base models. As shown in Table 1, this baseline achieved its highest accuracy of 33.99% on the ED dataset using the Mistral-Nemo model.

[4]https://huggingface.co/mistralai/Mistral-Nemo-Base-2407
[5]https://huggingface.co/microsoft/Phi-3.5-mini-instruct
[6]https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

Table 1: Baseline performance on Phi-3.5-mini, Mistral-Nemo, Llama3.1-8B models

Model	ED Dataset			EDOS Dataset		
	Accuracy	Distinct-1	Distinct-2	Accuracy	Distinct-1	Distinct-2
Phi-3.5-mini	33.38	0.037	0.274	29.56	0.068	0.362
Mistral-Nemo	33.99	0.044	0.226	27.31	0.071	0.286
Llama3.1-8B	32.69	0.020	0.104	20.47	0.025	0.091

4.3 Main Results

This section shows the main results of our research. First, we obtain the baseline performance. Then, we test our different enhancement methods against this baseline to find the best combination.

4.3.1 Overall Performance and Method Comparison

To systematically evaluate our approach, we compare the performance of three primary methods: COMET-Situation-based prompt augmentation, ConceptNet-based augmentation, and the Vector-Modulated Collaborative Prompting (VMCP) method. We run these experiments on the Mistral-Nemo model, and the results are in Table 2.

We use the Zero-Shot ICL framework as our baseline (row 1), achieving an accuracy of 33.99%. While only adding Situation knowledge (row 2) or the VMCP method (row 4), they both bring great improvement. In comparison, adding only ConceptNet knowledge (row 3) yields a slight improvement.

What we find interesting is that for the main ED dataset, adding all methods together (row 5, Initial Full Framework) performs worse than our final proposed combination (row 6, Our Proposed Best Model), which excludes the ConceptNet method. The best accuracy on ED is 45.96%. This shows that for the ED dataset, combining COMET-Situation and VMCP is the best approach, and adding ConceptNet hurts performance.

However, the results are different for the cross-domain EDOS dataset. On this dataset, the best performance comes from the Initial Full Framework, which includes ConceptNet, achieving an accuracy of 37.58%. This suggests that the usefulness of lexical knowledge like ConceptNet may depend on the specific dataset. Therefore, for our main experiments on ED, we define our best-performing model as Zero-Shot ICL + COMET-Situation + VMCP.

4.3.2 Cross-Model Generalization of Core Components

We found that +COMET-Situation and +VMCP were the best single methods. Here, we test if they work well on all our base models. The results in Table 3 show that both

Table 2: Main results and method comparison on **ED and EDOS datasets** with the **Mistral-Nemo model**. This table summarizes our primary methods and presents our optimal combination of solutions. D-1 and D-2 stands for Distinct-1/2

Method Configuration	**ED Dataset**			**EDOS Dataset**		
	Acc.	**D-1**	**D-2**	**Acc.**	**D-1**	**D-2**
ICL (Baseline)	33.99	0.044	0.226	27.31	0.071	0.286
Individual Method Contributions						
+COMET-Situation	40.08 ↑	0.045 ↑	0.226 ↓	27.43 ↑	0.073 ↑	0.285 ↓
+ ConceptNet	36.04 ↑	0.056 ↑	0.263 ↑	28.60 ↑	0.084 ↑	0.329 ↑
+ VMCP	40.25 ↑	0.043 ↓	0.224 ↓	37.46 ↑	0.070 ↓	0.283 ↓
Combined Frameworks						
Initial Full Framework [a]	44.89 ↑	0.053 ↑	0.254 ↑	**37.58** ↑	**0.094** ↑	**0.353** ↑
Our Proposed Best Model [b]	**45.96** ↑	**0.044** ↑	**0.225** ↓	36.01 ↑	0.069 ↓	0.282 ↓

[a] Initial Full Framework: Refers to the addition of baseline, COMET - Situation, ConceptNet, and VMCP methods.

[b] Our Proposed Best Model: Refers to the addition of **baseline, COMET - Situation, and VMCP method**.

'+COMET-Situation' and '+VMCP' consistently outperform the baseline. This confirms that both methods are effective on the in-domain dataset.

The results on the cross-domain EDOS dataset are more complex. The +VMCP method remains very effective, improving accuracy across all models. However, the +COMET-Situation method has mixed results. While it helps slightly on some models, it causes a large drop in accuracy for the Phi-3.5-mini model. This shows that this knowledge source does not generalize as well.

However, their effect on response diversity (as measured by the Distinct scores) varied for each model. The +COMET-Situation method generally increased or maintained the Distinct scores, suggesting that providing narrative context helps generate more diverse responses. In contrast, the +VMCP method almost always decreased the Distinct scores. This shows a clear trade-off. While the precise control of VMCP is good for accuracy, it seems to make the model's responses less diverse.

4.3.3 Performance of the Combined Model

Next, we checked if +COMET-Situation and +VMCP work even better when used together. The results in Table 4 show that for the main ED dataset, combining them gives the best performance. On Mistral-Nemo, the accuracy reached 45.96%, which was over 5% higher than using either method alone. The accuracy also jumped to 41.27% on Llama3.1-8B and 40.89% on Phi3.5-mini, making this combination the most effective.

However, it is different for the cross-domain EDOS dataset. For this dataset, combining

Table 3: Performance comparison of core components ('+COMET-Situation', '+VMCP') against the baseline across all three base models on both **ED and EDOS datasets**. D-1 and D-2 stands for Distinct-1/2

Base Model	**ED Dataset**			**EDOS Dataset**		
	Baseline	**+Situation**	**+VMCP**	**Baseline**	**+Situation**	**+VMCP**
Phi-3.5-mini	Acc: 33.38 D-1: 0.037 D-2: 0.274	Acc: 37.70 ↑ D-1: 0.038 D-2: 0.283	Acc: 37.11 ↑ D-1: 0.036 D-2: 0.270	Acc: 29.56 D-1: 0.068 D-2: 0.361	Acc: 19.35 ↓ D-1: 0.088 ↑ D-2: 0.423 ↑	Acc: 35.11 ↑ D-1: 0.068 ↑ D-2: 0.360 ↓
Mistral-Nemo	Acc: 33.99 D-1: 0.044 D-2: 0.226	Acc: 40.08 ↑ D-1: 0.045 ↑ D-2: 0.226 ↓	Acc: 40.25 ↑ D-1: 0.043 ↓ D-2: 0.224 ↓	Acc: 27.31 D-1: 0.071 D-2: 0.286	Acc: 27.43 ↑ D-1: 0.073 ↑ D-2: 0.285 ↓	Acc: 37.46 ↑ D-1: 0.070 ↓ D-2: 0.283 ↓
Llama3.1-8B	Acc: 29.91 D-1: 0.019 D-2: 0.095	Acc: 33.83 ↑ D-1: 0.018 ↓ D-2: 0.096 ↑	Acc: 37.62 ↑ D-1: 0.016 D-2: 0.085	Acc: 20.47 D-1: 0.025 D-2: 0.091	Acc: 22.66 ↑ D-1: 0.026 ↑ D-2: 0.096 ↑	Acc: 30.12 ↑ D-1: 0.024 ↓ D-2: 0.088 ↓

Table 4: Main evaluation results of the combined model ('+COMET-Situation+VMCP') on both **ED and EDOS datasets**. D-1 and D-2 stands for Distinct-1/2

Base Model	**ED Dataset**			**EDOS Dataset**		
	+Situation	**+VMCP**	**+Combo**	**+Situation**	**+VMCP**	**+Combo**
Phi3.5-mini	Acc: 37.70 D-1: 0.038 D-2: 0.283	Acc: 37.11 D-1: 0.036 D-2: 0.270	**Acc: 40.89** ↑ D-1: 0.039 ↑ D-2: 0.283 ↑	Acc: 19.35 D-1: 0.088 D-2: 0.423	Acc: 35.11 D-1: 0.068 D-2: 0.360	Acc: 21.54 ↓ D-1: 0.089 ↑ D-2: 0.427 ↑
Mistral-Nemo	Acc: 40.08 D-1: 0.045 D-2: 0.226	Acc: 40.25 D-1: 0.043 D-2: 0.224	**Acc: 45.96** ↑ D-1: 0.044 ↓ D-2: 0.225 ↓	Acc: 27.43 D-1: 0.073 D-2: 0.285	Acc: 37.46 D-1: 0.070 D-2: 0.283	Acc: 36.01 ↓ D-1: 0.069 ↓ D-2: 0.282 ↓
Llama3.1-8B	Acc: 33.83 D-1: 0.018 D-2: 0.096	Acc: 37.62 D-1: 0.016 D-2: 0.085	**Acc: 41.27** ↑ D-1: 0.017 ↓ D-2: 0.089 ↓	Acc: 22.66 D-1: 0.026 D-2: 0.096	Acc: 30.12 D-1: 0.024 D-2: 0.088	Acc: 29.95 ↓ D-1: 0.028 ↑ D-2: 0.105 ↑

the two methods is not the best strategy. The table shows that for all three base models, the '+VMCP' method used alone has a higher accuracy than the combined ('+Combo') model. This result aligns with our earlier finding that the '+COMET-Situation' knowledge, which is based on the ED dataset, does not generalize well and can even hurt performance when added in a cross-domain context.

The effect on response diversity also depended on the model and dataset. On the ed dataset, for the Phi3.5-mini, the diversity score increased. However, for Mistral-Nemo and Llama3.1-8B, the large improvement in accuracy came with a small drop in the diversity scores.

Table 5: The performance of different prompt enhancement strategies with **ED** dataset on the **Mistral-Nemo model**. The narrative-based **+COMET-Situation** strategy performed the best.

Strategy	**Accuracy**	**Dist-1**	**Dist-2**
Baseline			
ICL (Mistral-Nemo)	33.99	0.044	0.226
+ Prompt Augmentation (Lexical Features)			
+ VAD	34.58 ↑	0.046 ↑	0.231 ↑
+ ConceptNet	**36.04** ↑	0.056 ↑	0.263 ↑
+ VAD & ConceptNet	35.26 ↑	**0.074** ↑	**0.307** ↑
+ Prompt Augmentation (Narrative Knowledge)			
+ COMET-Situation	**40.08** ↑	**0.045** ↑	**0.226** ↑
+ COMET-xReact	33.49 ↓	0.039 ↓	0.211 ↓
+ COMET-xIntent	33.51 ↓	0.040 ↓	0.215 ↓

4.4 Analysis of Design Choices

This section provides more analysis to justify our key design choices.

4.4.1 Evaluation of Prompt-based Augmentation Strategies

To justify selecting ‘COMET-Situation‘ as our primary prompt augmentation strategy, we analyzed the performance of different knowledge sources on our main model, Mistral-Nemo. As shown in Table 5, ‘+COMET-Situation‘ shows the best accuracy improvement.

To ensure this finding was not model-specific, we verified its generalizability across our other base models. As the results in Table 11 and Table 12 show, ‘+COMET-Situation‘ always shows better accuracy.

In contrast, all our tested lexical-based strategies, like VAD, ConceptNet, and COMET’s lexical outputs, proved to be unreliable and highly model-dependent. For instance, even on the best-performing model, Mistral-Nemo, the results were mixed: while methods like ‘+VAD‘ and ‘+ConceptNet‘ offered a small accuracy increase, others such as ‘+COMET-xReact‘ performed worse than the baseline (Table 5). Crucially, this inconsistency was magnified on the Llama3.1-8B and Phi-3.5-mini models, where all tested lexical strategies failed to outperform the baseline (see Table 11, and Table 12). We analyze the potential reasons for this general underperformance in Chapter 5.

Given the strong, consistent performance of ‘+COMET-Situation‘ versus the unreliable results of all lexical-based methods, we selected ‘+COMET-Situation‘ as the best prompt augmentation method for our final proposed framework.

Table 6: Performance comparison of combining ConceptNet and COMET-Situation knowledge. D-1 and D-2 stands for Distinct-1/2

Base Model	+ConceptNet	+COMET-Situation	+ConceptNet+Situation
Phi3.5-mini	Acc: 32.45 D-1: 0.037 D-2: 0.286	Acc: 37.70 D-1: 0.038 D-2: 0.283	Acc: 37.30 ↓ **D-1: 0.039** ↑ **D-2: 0.293** ↑
Mistral-Nemo	Acc: 36.04 D-1: 0.056 D-2: 0.263	Acc: 40.08 D-1: 0.045 D-2: 0.226	**Acc: 40.36** ↑ D-1: 0.053 ↓ D-2: 0.255 ↓
Llama3.1-8B	Acc: 29.69 D-1: 0.018 D-2: 0.095	Acc: 33.83 D-1: 0.018 D-2: 0.096	Acc: 31.93 ↓ D-1: 0.017 ↓ D-2: 0.093 ↓

4.4.2 Details of Design Choices for External Knowledge

After finding that the COMET-Situation and VAD methods were promising, we conducted further studies and parameter analyses to investigate the impact of different design choices.

First, we explore the interaction between the two primary knowledge sources, COMET-Situation and ConceptNet. As shown in Table 6, combining the two methods only outperforms each method alone on the Mistral-Nemo model. On other models, it even degraded performance. What we find interesting is that on the Phi-3.5-mini model, the Distince-1/2 has a rise in performance.

Secondly, we analyzed the design choices for the VAD keywords, including the extraction range and the location within the prompt. Our research results indicate that focusing on the last speaker's utterances and placing the keywords after the "Dialogue Context" will lead to better results. The detailed results of these parameter analyses are listed in Appendix A.

4.4.3 Ablation Study of VMCP Components

Finally, to validate the design of our core VMCP method, we conducted an ablation study on its two-stage mechanism. As shown in Table 7, the full VMCP method (EGI + AIR) significantly outperforms the baseline. Surprisingly, relying solely on the initial External Guidance Injection (EGI only) yields a performance of 40.27%, higher than the full two-stage method. This result suggests that while our two-stage mechanism is effective, the initial guidance injection (EGI) is the primary reason for the performance gain. The Adaptive Internal Refinement (AIR) stage, in its current implementation, does not provide an additional benefit and may even introduce a slight amount of noise.

Table 7: Ablation study of the core components of our VMCP method (EGI and AIR) on the ED dataset, using Mistral-Nemo as the base model.

Method Configuration	**Accuracy**	**Distinct-1**	**Distinct-2**
ICL + VMCP (Full: EGI + AIR)	**40.25**	0.043	0.224
w/o AIR (EGI only)	40.27	0.043	0.225
ICL (Baseline)	33.99	0.044	0.226

4.5 Qualitative Analysis

To provide a more intuitive understanding of our methods, we present three case studies using the Mistral-Nemo model.

Case 1: 'Lonely' Emotion (Table 8) In the breakup scenario, the baseline response is generic and supportive. The +COMET-Situation method is better because it uses a specific detail from the dialogue ("after being together for so long"). Our final combined model (+Situation+VMCP) keeps this specific detail but also asks a more helpful, proactive question ("any support you need right now?").

Case 2: 'Trusting' Emotion (Table 9) This case shows an interesting result. While all methods give good, encouraging responses, the most complex model (+Situation+VMCP) gives one of the simplest replies ("That's wonderful! It's great to trust our instincts."). This shows that for simple, positive emotions, our combined method can make the response more direct instead of more complex.

Case 3: 'Angry' Emotion (Table 10) This example shows the limits of our VMCP method. The table shows that the +VMCP response is the same as the baseline. Similarly, the final combined model's response is the same as using +COMET-Situation by itself. This tells us that when the dialogue already has an obvious emotional signal (like "angry"), the extra guidance from VMCP doesn't change the final output.

Table 8: Comparison of generated responses for the 'Lonely' emotion, based on different knowledge sources; the model used is Mistral-Nemo. Abbreviations are in the notes below.

Emotion	Lonely
Dialogue Context	Speaker: I'm there, don't know what to do, just broke up with my girlfriend, we were 8 years together Listener: Sorry to hear! Do you have any idea about the breakup? Did you think about it ? Speaker: Yes, we decided together with our minds, and now I come home and feel so distant from the world
External Knowledge	COMET-Situation: The speaker just broke up with the speaker's girlfriend , they were together for 8 years , the speaker feel so empty . ConceptNet: Feel is related to regret; World is related to dreaming; Home is related to guardian; Distant is related to emotionally; Together is related to mate VAD: Feel; World; Home; Distant; Together
Generated Responses from Different Methods	
ICL (Baseline)	I'm really sorry for your loss. It's completely normal to feel this way after such a long relationship. Take all the time you need to heal.
+COMET-Situation	I'm really sorry to hear that. It must be incredibly difficult after being together for so long. Is there anything you'd like to talk about or need help with?
+ConceptNet	I'm really sorry that you're going through this. It's completely normal to feel lost and distant after such a long relationship. Is there anyone else you can talk to or spend time with at home?
+VMCP	I'm really sorry that you're feeling this way. It's completely normal to feel lost after such a long relationship. Would you like to talk about how you're feeling?
+COMET-Situation+VMCP	I'm really sorry to hear that. It must be incredibly difficult after being together for so long. Is there anything you'd like to talk about or any support you need right now?
Prompt	See the prompt structure in Chapter 4.1 ('Augmented Prompt Examples')

Table 9: Comparison of generated responses for the 'Trusting' emotion, based on different knowledge sources; the model used is Mistral-Nemo. Abbreviations are in the notes below.

Emotion	Trusting
Dialogue Context	Speaker: i am a big believer in listening to your gut . sometimes i might wonder where my intuition is leading me , but it always seems to be spot on . Listener: i agree , someimes we just know Speaker: it is a knowing – and a flowing ! when something is right , it just feels easier to go with it – like you are being led , even if you might not fully know to where .
External Knowledge	COMET-Situation: The speaker's intuition has led the speaker into some situations that had the speaker only used the speaker's head , the speaker might have missed out on . the speaker am a big believer in listening to your gut . ConceptNet: Believer is related to believe; Like is related to habit; Knowing is a type of believing; Wonder is related to sense; Intuition is related to feeling
Generated Responses from Different Methods	
ICL (Baseline)	That's a great way to put it! Trust in yourself.
+COMET-Situation	That's great! Trusting your instincts can lead to amazing experiences.
+ConceptNet	That's great to hear! Trusting your instincts can lead to amazing outcomes. Keep up that confidence!
+VMCP	That's beautifully put. It's amazing how our instincts can guide us.
+COMET-Situation+VMCP	That's wonderful! It's great to trust our instincts.
Prompt	See the prompt structure in Chapter 4.1 ('Augmented Prompt Examples')

Table 10: Comparison of generated responses for the 'Angry' emotion, based on different knowledge sources; the model used is Mistral-Nemo. Abbreviations are in the notes below.

Emotion	Angry
Dialogue Context	Speaker: i recently had a big argument with my sister . we both got very upset and angry at each other . Listener: is not that what sisters are for ? what were you guys upset about ? Speaker: i am not so sure . we just had a discussion that heated and became an argument with tempers flying .
External Knowledge	COMET-Situation: The speaker recently got very upset with the speaker's sister when said some very mean things to the speaker . the speaker let her know that she was out of line by yelling at her . the speaker feel bad for both of them . ConceptNet: Argument is related to clash; Sister is a type of person; Upset is related to upheaval; Discussion is a type of argument; Big is related to crowd
Generated Responses from Different Methods	
ICL (Baseline)	I'm sorry to hear that. Arguments can be tough, but it's important to talk things out when you're calm. What was the argument about?
+COMET-Situation	I'm sorry to hear that. It's never easy when family members argue. What did your sister say that made you so upset?
+ConceptNet	I see. It's never easy when arguments get out of hand, especially with family. What was the main topic that led to this?
+VMCP	I'm sorry to hear that. Arguments can be tough, but it's important to talk things out when you're calm. What was the argument about?
+COMET-Situation+VMCP	I'm sorry to hear that. It's never easy when family members argue. What did your sister say that made you so upset?
Prompt	See the prompt structure in Chapter 4.1 ('Augmented Prompt Examples')

Chapter 5 Discussion

In this study, we evaluated the two optimization methods: external prompt augmentation and internal VMCP intervention. We build the discussion on top of the experiment results in Chapter 4. We want to go beyond just performance numbers and talk about the mechanisms behind them, discuss the meaning of our core contributions, and explore future research directions.

5.1 Explanation of Prompt-based Augmentation Strategies

5.1.1 The 'Role Mismatch' Challenge in COMET Knowledge

All knowledge from COMET (including Situation, xReact, and xIntent) has a common risk we call 'Role Mismatch'. COMET generates knowledge from the perspective of the event's experiencer. However, our task is to predict the emotion of the last speaker, who is not always the same person.

To prove the 'role mismatch' that we are facing, here is an example from the ED test data:

> **Speaker 1 (Event Experiencer):** "...i wanted to tell you about the time i was hit by a drunk driver im so happy to still be alive after that experience."
> **Speaker 2 (Last Speaker):** "oh my goodness, that is very scary! i hope you are okay now and the drunk driver was punished for his actions?"

The ground truth emotion for this interaction, focusing on **Speaker 2**, is **'caring'**. However, the commonsense knowledge generated from the core event would relate to Speaker 1's experience. This external knowledge about Speaker 1's state is irrelevant and potentially misleading for predicting Speaker 2's 'caring' emotion.

To solve this problem, we created a filter based on semantic similarity. The filter compares the COMET-Situation text to the last speaker's text. We only add the COMET knowledge to the prompt if the cosine similarity score is above 0.5. We do this because a high score suggests that the last speaker is also the person who experienced the event, making the knowledge relevant.

Interestingly, our results show that even with this strict filter, only the narrative situation knowledge improved the model's performance, as is shown in Table 5. Adding other types of knowledge that also passed the filter, like the lexical xReact, still did not work well.

5.1.2 Comparative Analysis of Narrative Knowledge vs. Lexical Features

From the big difference in results on adding external knowledge from Situation and xReact, we analyze the reasons. A key finding is the large performance difference between narrative knowledge (COMET-Situation) and lexical features (COMET-xReact, VAD keywords). We think this indicates that different knowledge types have different adaptability in the ICL scenario, in which a long context, like the COMET-Situation, gives more useful information during the model reasoning process. While in a fine-tuning scenario, strong signal features like xReact will be valuable because the model can learn to adjust its weights. However, in our ICL setting, model weights are frozen, so the model cannot "learn" to adapt to these new features.

In this ICL scenario, the flaws of xReact are more obvious. First, there is the risk of "over-spoiling": giving the model label-like emotional words may short-circuit its deep reasoning process. Second, there can be an "expert conflict": the judgment from the small-expert COMET may conflict with the internal judgment of the large-language model.

In contrast, the advantages of situation knowledge are fully realized in the ICL scenario. It provides a neutral, complete background story, which LLMs are good at understanding from their pre-training. Therefore, we conclude that in the weight-frozen ICL scenario, giving the model a narrative context is a more effective strategy than giving it signal-like features that require the model to "learn" how to use them.

5.1.3 Analysis of VAD and ConceptNet

Unlike the general improvement from the COMET-Situation strategy, methods that use isolated lexical features, like VAD and ConceptNet, also show unstable effects in our experiments.

We find that the performance of these lexical strategies depends heavily on the model we use. Specifically, '+VAD', '+ConceptNet', and their combination improve accuracy only on Mistral-Nemo (Table 5), but decrease accuracy on both Llama3.1-8B (Table 11) and Phi-3.5-mini (Table 12). This suggests that different models have different abilities to use multiple information sources in a zero-shot setting. We hypothesize that Mistral-Nemo is better at using these lexical cues as hints, while other models treat them as distracting noise that interferes with their reasoning. The following sections analyze the specific problems with these features that cause this result.

1. Analysis of VAD Limitations

Our analysis of the '+VAD' strategy shows it is not a consistently effective method. As seen in our experiments, it provides only a small accuracy increase on Mistral-Nemo and fails to improve performance on Llama3.1-8B and Phi-3.5-mini.

We identify the core reason for this poor performance: VAD extracts keywords based on

predefined emotional scores, without understanding their surrounding context. A clear example is in the phrase "I am not sad", where VAD still extracts "sad" as a high-value emotional keyword because it ignores the word "not".

This context-blindness means the strategy provides the model with a noisy and often misleading signal, and it helps explain the model-dependent results we observe. We hypothesize that a more reliable model like Mistral-Nemo can better handle or filter this noise, leading to the small performance gain. In contrast, other models like Llama3.1-8B and Phi-3.5-mini appear to be misled by the incorrect keywords, which hurts their accuracy.

2. Analysis of VAD Extraction Scope

A key design choice was the source for VAD keyword extraction. We tested two approaches: using the full dialogue history (VAD-old) and focusing only on the last speaker's utterance (VAD-new). We hypothesize that VAD-new would be more precise and reduce noise from the other speaker's context.

Our experimental results confirm this hypothesis. As the data in Appendix Appendix A.2 shows, the VAD-new method consistently achieved higher accuracy. In other words, using the full dialogue (VAD-old) adds confusing words from the other speaker. By focusing solely on the last speaker, we ensure the keywords are a good match for the target emotion.

Therefore, to ensure higher accuracy, we adopted VAD-new as the standard method for our VAD-based prompt enhancements.

3. Analysis of the Location of the VAD external knowledge

The results are significantly improved when the VAD words are placed after the dialogue context, as shown in Table 15, with an accuracy increase from 33.89% to 34.58%.

There is a simple reason behind this result: when the dialogue comes first, it allows the model to understand the whole story. Then the VAD words come as a hint to guide the model when making final decisions.

However, if the VAD words come first, the model receives these words before seeing the entire story, which might confuse the model and lead to wrong guesses.

Therefore, placing the dialogue first is the correct approach. It gives the model the main story before it sees the final hints.

4. Analysis of ConceptNet Limitations

We also test the effect of adding general semantic relations from ConceptNet on top of the VAD keywords. Our VAD method is good at finding key emotional words, but it doesn't capture the relationships between concepts. We tested ConceptNet to see if it could add this missing knowledge.

This combination produced interesting results, as shown in Table 5, for the Mistral-Nemo model, it worked well. The relational knowledge from ConceptNet appeared to comple-

ment the VAD keywords, resulting in a clear performance improvement.

However, this improvement was unique to Mistral-Nemo. In our other models (Table 11 and Table 12), adding ConceptNet's knowledge hurt performance.

A possible reason for this is that models have different abilities to handle complex information. Mistral-Nemo may be better at using the ConceptNet relations as a helpful "hint" to enrich the VAD keywords. In contrast, the other models likely treated this general knowledge as distracting "noise" rather than a valuable knowledge source, which interfered with the reasoning process.

Our findings show a strong model-dependency for these lexical methods. The fact that +ConceptNet helps Mistral-Nemo but hurts the other models is an important finding. It shows that not all models are good at using this kind of sparse, keyword-based knowledge in a zero-shot setting. Mistral-Nemo seems better at using these keywords as helpful 'hints', while the other models seem to treat them as 'noise' that disrupts their reasoning.

5.2 Core Mechanism and Deeper Impact of VMCP

Our results in Chapter 4 show interesting conclusions about VMCP. It works as well as our best prompt method. And when we combine them, the results are much better. This shows that the two methods help each other and do different jobs.

5.2.1 VMCP's Performance Compared to Prompting

The key result from our experiments is surprising on the ED dataset. As shown in Table 3, the +VMCP method is very powerful. On the Phi-3.5-mini and Mistral-Nemo models, its performance is nearly the same as our best prompt method, +COMET-Situation. On the Llama3.1-8B model, +VMCP is even better. This shows that directly guiding the model's internal state can be just as effective as giving it a rich story in the prompt.

The two methods take different paths to the same goal. The +COMET-Situation prompt gives the model a background story. The model then reasons about this story to find the emotion. VMCP is more direct. It "pushes" the model's internal state toward the right emotion. This direct push is a very efficient way to get the correct result.

5.2.2 Combining External Prompts and Internal Control

We get the best results when we combine the prompt augmentation method (COMET-Situation) and vector injection method (VMCP). The +COMET-Situation+VMCP model performs much better than either method used alone, as shown in Table 4. This shows that the two methods work well together.

They do different jobs. The COMET-Situation prompt provides the "why" by giving the model a background story for the emotion. VMCP provides the "how" by giving the model a direct signal to create that emotion. Using both the story context and the direct control signal leads to the best performance.

5.2.3 Analysis of Accuracy vs. Diversity

We also noticed a trade-off in our results. VMCP does a great job on accuracy, but it sometimes lowers the diversity of the generated responses (the Distinct-1/2 scores), as shown in Table 3.

This may be a natural side effect of VMCP's "anchoring" job. It strongly guides the model toward a specific emotional state. This might reduce the variety of words and phrases the model uses. In contrast, giving the model a story with +COMET-Situation allows more freedom in how it replies, which leads to more diverse answers. When combining the two methods, for the ED dataset, the Distinct-1/2 number increases on the Phi-3.5-mini model, as shown in Table 4, while on the other models, it slightly lowers performance.

This result shows a trade-off between precise control of emotion generation and response diversity. It is an interesting area for future work to find a balance between both.

This trade-off may show a basic conflict between control and creativity. Our VMCP method is very good at forcing the model toward a specific emotion to improve accuracy. But this tight control leaves less room for the model to be creative, which is why response diversity goes down. A good direction for future work is to find a better balance. We could develop methods with "softer" guidance that ensure emotional accuracy but still give the model more creative freedom.

5.2.4 The Critical Role of Adaptive Internal Refinement (AIR)

Our ablation study in Table 7 shows a surprising result. When we remove the AIR stage and use only the initial EGI stage, the performance is actually slightly better, increasing from 40.25% to 40.27%. This tells us two things about our two-stage design:

1. **EGI is the most important part.** The initial vector injection from EGI is what provides the main performance boost over the baseline.

2. **The AIR stage is not helpful in our setup.** AIR was designed to fine-tune the result, but the data shows it doesn't add any benefit. It might even add a small amount of noise or "over-correction" that slightly hurts the performance.

So, while our two-stage design is effective, our results show that the power of VMCP comes almost entirely from the initial EGI step. A clear direction for future work is to explore different refinement methods for the second stage that could help improve the result.

5.3 Generalization to Cross-Domain Data

Our cross-domain tests on the EDOS dataset gave us some interesting results. As shown in Table 2, how well the prompt-based knowledge augmentation works depends a lot on the knowledge source. The +COMET-Situation method, which worked great on the ED dataset (a +6.09 point accuracy gain), barely helped on the EDOS dataset (only a +0.12 point gain).

In contrast, our internal control method, VMCP, worked well on both datasets. Most importantly, on the EDOS dataset (Table 4), VMCP used alone was better than the combined +Situation+VMCP model (37.46% vs. 36.01% accuracy).

This result shows that in a cross-domain setting, adding low-quality or "noisy" external knowledge can mess up an already effective internal guidance signal. It also shows that VMCP is a reliable method even when the conditions are not perfect.

5.4 Limitations and Future Work

Our study has some weaknesses despite its positive results. Our main weakness is the metric we used. We used "emotion recognition accuracy" as our primary metric to evaluate performance.

This metric was a fast and helpful way to measure our models. However, this score does not directly check the quality of the final written response.

Therefore, a key part of our future work will be to use large-scale human evaluation. We need to have many people read and rate the responses our model generates. Humans can grade important qualities like fluency and the level of empathy. This extension will enable us to fully demonstrate the effectiveness of our combined method in improving the final response quality.

Chapter 6 Conclusion

In this paper, we show how to generate more empathetic responses from large language models using In-Context Learning. We test two methods: first, we augment input prompts with external knowledge, and second, we directly guide the model's internal reasoning with our proposed Vector-Modulated Collaborative Prompting (VMCP) method.

Our experiments show three main results. First, we show that for prompt augmentation, narrative knowledge (COMET-Situation) is more effective than lexical features (VAD, ConceptNet, or COMET's lexical outputs). We conclude that in a zero-shot ICL setting, providing full contextual stories is more effective than adding lexical keywords. Second, we propose VMCP, a two-stage method that uses emotion vector injection and addition. Our results show that VMCP is an effective method for guiding the model's emotion, with performance similar to or better than the best prompt augmentation strategy.

While many existing works separately use prompt design and internal control, our results show that their combination can be very effective. By adding external narrative context with internal vector guidance, we provide both narrative context (the "why") and direct emotional steering (the "how") in LLMs, which gives the best performance for the task.

Additionally, our cross-domain tests show the limits of our hybrid model, indicating that in settings where external knowledge does not generalize well, our internal guidance method (VMCP) is more reliable on its own.

For future work, we note a limitation. Our findings are based on automatic metrics for emotion classification accuracy. These metrics do not fully measure the quality of an empathetic response. Therefore, a key next step is to conduct large-scale human evaluations. This is necessary to validate that the higher accuracy also leads to responses that are more fluent, appropriate, and genuinely empathetic.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Emmanuelle Zech and Bernard Rimé. Is talking about an emotional experience helpful? effects on emotional recovery and perceived benefits. *Clinical Psychology & Psychotherapy: An International Journal of Theory & Practice*, 12(4):270–287, 2005.

[3] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv:1811.00207*, 2018.

[4] Yushan Qian, Wei-Nan Zhang, and Ting Liu. Harnessing the power of large language models for empathetic response generation: Empirical investigations and improvements. *arXiv preprint arXiv:2310.05140*, 2023.

[5] Sahand Sabour, Chujie Zheng, and Minlie Huang. Cem: Commonsense-aware empathetic response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11229–11237, 2022.

[6] Qintong Li, Piji Li, Zhaochun Ren, Pengjie Ren, and Zhumin Chen. Knowledge bridging for empathetic dialogue generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 10993–11001, 2022.

[7] Zhou Yang, Zhaochun Ren, Yufeng Wang, Chao Chen, Haizhou Sun, Xiaofei Zhu, and Xiangwen Liao. An iterative associative memory model for empathetic response generation. *arXiv preprint arXiv:2402.17959*, 2024.

[8] Hua Cai, Xuli Shen, Qing Xu, Weilin Shen, Xiaomei Wang, Weifeng Ge, Xiaoqing Zheng, and Xiangyang Xue. Improving empathetic dialogue generation by dynamically infusing commonsense knowledge. *arXiv preprint arXiv:2306.04657*, 2023.

[9] Zhaojiang Lin, Andrea Madotto, Jamin Shin, Peng Xu, and Pascale Fung. Moel: Mixture of empathetic listeners. *arXiv preprint arXiv:1908.07687*, 2019.

[10] Navonil Majumder, Pengfei Hong, Shanshan Peng, Jiankun Lu, Deepanway Ghosal, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. Mime: Mimicking emotions for empathetic response generation. *arXiv preprint arXiv:2010.01454*, 2020.

[11] Qintong Li, Hongshen Chen, Zhaochun Ren, Pengjie Ren, Zhaopeng Tu, and Zhumin Chen. Empdg: Multiresolution interactive empathetic dialogue generation. *arXiv preprint arXiv:1911.08698*, 2019.

[12] Anonymous Author(s). How ICL Makes Decisions in Fine-Grained Emotion Recognition: A Prototype Perspective. Unpublished manuscript, Leiden University, 2025, 2025.

[13] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035, 2019.

[14] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*, 2019.

[15] Jinfeng Zhou, Chujie Zheng, Bo Wang, Zheng Zhang, and Minlie Huang. Case: Aligning coarse-to-fine cognition and affection for empathetic response generation. *arXiv preprint arXiv:2208.08845*, 2022.

[16] Saif Mohammad. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 174–184, 2018.

[17] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[18] Zhou Yang, Zhaochun Ren, Yufeng Wang, Xiaofei Zhu, Zhihao Chen, Tiecheng Cai, Yunbing Wu, Yisong Su, Sibo Ju, and Xiangwen Liao. Exploiting emotion-semantic correlations for empathetic response generation. *arXiv preprint arXiv:2402.17437*, 2024.

[19] Weixiang Zhao, Yanyan Zhao, Xin Lu, and Bing Qin. Don't lose yourself! empathetic response generation via explicit self-other awareness. *arXiv preprint arXiv:2210.03884*, 2022.

[20] Guanqun Bi, Lei Shen, Yanan Cao, Meng Chen, Yuqiang Xie, Zheng Lin, and Xiaodong He. Diffusemp: A diffusion model-based framework with multi-grained control for empathetic response generation. *arXiv preprint arXiv:2306.01657*, 2023.

[21] Anuradha Welivita, Yubo Xie, and Pearl Pu. A large-scale dataset for empathetic response generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1251–1264, 2021.

Appendix A Supplementary Experimental Details

The appendix provides the results and detailed analyses of the key designs and hyperparameter settings mentioned in the main text.

A.1 Detailed Prompt Augmentation Results on All Models

To complete the analysis in Section 4.4.1, this section records the results of all the prompt enhancement strategies tested on the Llama3.1-8B and Phi-3.5-mini models in detail. As shown in Table 11 and Table 12, the narrative-based +COMET-Situation strategy is consistently the most effective method across different model architectures.

On the other hand, the +ConceptNet method was not reliable. It even hurt performance on two of the models, which is why we extracted it from our final method.

Table 11: The performance of different prompt enhancement strategies with the **ED** dataset on the **Llama3.1-8B model**. The narrative-based **+COMET-Situation** strategy performed the best.

Strategy	**Accuracy**	**Dist-1**	**Dist-2**
Baseline			
ICL (Llama3.1-8B)	29.91	0.019	0.095
+ Prompt Augmentation (Lexical Features)			
+ VAD	28.83 ↓	**0.033** ↑	**0.179** ↑
+ ConceptNet	29.69 ↓	0.018 ↓	**0.095** ↑
+ VAD & ConceptNet	28.68 ↓	0.018 ↓	0.092 ↓
+ Prompt Augmentation (Narrative Knowledge)			
+ COMET-Situation	**33.83** ↑	0.018 ↓	**0.096** ↑
+ COMET-xReact	28.92 ↓	**0.026** ↑	**0.141** ↑
+ COMET-xIntent	27.76 ↓	**0.023** ↑	**0.121** ↑

A.2 VAD Keyword Selection

We explored several parameters for extracting VAD keywords.

Extraction Scope. We compared extracting keywords from the full dialogue context ('VAD-old') with only the last speaker's utterances ('VAD-new'). On Mistral-Nemo, 'VAD-

Table 12: The performance of different prompt enhancement strategies with the **ED** dataset on the **Phi-3.5-mini model**. The narrative-based **+COMET-Situation** strategy performed the best.

Strategy	**Accuracy**	**Dist-1**	**Dist-2**
Baseline			
ICL (Phi-3.5-mini)	33.38	0.037	0.274
+ Prompt Augmentation (Lexical Features)			
+ VAD	32.79 ↓	0.036 ↓	0.270 ↓
+ ConceptNet	32.45 ↓	**0.037** ↑	**0.286** ↑
+ VAD & ConceptNet	31.51 ↓	0.036 ↓	**0.282** ↑
+ Prompt Augmentation (Narrative Knowledge)			
+ COMET-Situation	**37.70** ↑	**0.038** ↑	**0.283** ↑
+ COMET-xReact	29.04 ↓	0.035 ↑	0.265 ↓
+ COMET-xIntent	30.90 ↓	**0.038** ↑	0.289 ↓

new‘ performed slightly better, as shown in Table 13. This is the reason we use VAD-new during the experiment, as it outperforms VAD-old.

Table 13: Performance comparison of VAD extraction scopes on Mistral-Nemo.

VAD method	**Accuracy**	**Distinct-1**	**Distinct-2**
VAD old	34.25	0.045	0.230
VAD new	**34.58**	**0.046**	**0.231**

Number of Keywords (k). We tested different numbers of keywords to inject. As shown in Table 14, selecting the top 5 keywords offered the best performance on Mistral-Nemo. Using fewer keywords failed to capture the full emotional context, while using more introduced noise.

Table 14: Performance vs. number of VAD keywords (k) on Mistral-Nemo.

k Value	**Accuracy (%)**
4	34.50
5	**34.58**
6	34.56

Knowledge Location in Prompt. We found that placing VAD keywords *after* the dialogue context shows much better results than placing them before, as this allows the model to first understand the context before considering the keywords as hints. As Table 15 shows, on Mistral-Nemo, this change improved accuracy from 33.89% to 34.58%.

Table 15: Performance comparison of placing VAD external knowledge before versus after the dialogue context. This experiment was conducted on the ED dataset using the **Mistral-Nemo** model.

VAD Location in Prompt	Accuracy	Distinct-1	Distinct-2
Before Dialogue Context	33.89	0.054	0.250
After Dialogue Context	**34.58**	**0.046**	**0.231**

A.3 COMET Knowledge Filtering

To select the best COMET knowledge source for the ED dataset, we compared the coverage of narrative Situation knowledge against lexical xReact knowledge.

Using a relevance threshold of 0.5 for 'COMET-Situation', we could inject relevant context into a high proportion of the test samples, achieving a 91.4% injection rate (4,805 out of 5,255 samples). In contrast, for the lexical 'xReact' knowledge, even with a more permissive (lower) threshold of 0.28, the injection rate was only 29.8% (1,566 samples).

Because of its much higher coverage, we chose COMET-Situation as our primary knowledge source for our main experiments.

For the cross-domain EDOS dataset, which uses our direct retrieval method, we also selected a threshold of **0.5**. This value was chosen as it provided a good balance between finding a sufficient number of relevant matches (a 22.0% injection rate) and maintaining high quality, based on our preliminary tests and qualitative checks.