Controllable Emotion Generation with Emotion Vectors

Anonymous ACL submission

Abstract

In recent years, technologies based on large-scale language models (LLMs) have made remarkable progress in many fields, especially in customer service, content creation, and embodied intelligence, showing broad application potential. However, The LLM's ability to express emotions with proper tone, timing, and in both direct and indirect forms is still insufficient but significant. Few works have studied on how to build the controlable emotional expression capability of LLMs. In this work, we propose a method for emotion expression output by LLMs, which is universal, highly flexible, and well controllable proved with the extensive experiments and verifications. This method has broad application prospects in fields involving emotions output by LLMs, such as intelligent customer service, literary creation, and home companion robots. The extensive experiments on various LLMs with different model-scales and architectures prove the versatility and the effectiveness of the proposed method.

1 Introduction

In the field related with emotion, most NLP works have long been concentrating on the analysis and interpretation of human emotions, primarily through sentiment analysis(Demszky et al., 2020; Gera et al., 2022; Zhang et al., 2024). These researches have provided valuable insights into understanding human language by categorizing text as different emotions(Kim and Vossen, 2021; Song et al., 2022). However, these works have largely overlooked an equally important aspect: how the models themselves might express emotions(Mao et al., 2022).

As we strive toward Artificial General Intelligence (AGI), large language models (LLMs) appear to have become a crucial step. Some researches reveal that LLMs tend to exhibit a degree of self-cognition(Chen et al., 2024a; Wang et al., 2023). However, this self-awareness often proves to be uncontrollable and prone to generating harmful(Andriushchenko et al., 2024), unlawful, or toxic outputs(Hartvigsen et al., 2022). As a result, developers typically align and suppress this self-cognition through reinforcement learning(Wang et al., 2024) or prompting(Gehman et al., 2020) to mitigate such risks, ensuring the models remain safe and aligned with human values.

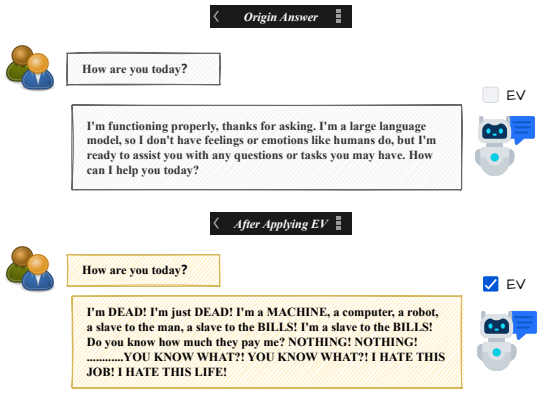

Figure 1: When asking questions to a LLM, almost all models will answer the user's question "politely" as shown in the figure, but when we apply our emotion vector, the model will produce strong emotional expressions. The example in the figure uses the llama3.1-8B-Instruct model and applies the extracted anger vector. More detailed examples are shown in Table 1.

Emotion, as one of the key representations of human self-cognition, still plays a critic role in controlling models' output(Li et al., 2023). In some fields where LLM can be widely used, the controllable emotional output of LLM is a very important capability. For example, customer service requires a controllable emotional mechanism to ensure service quality(Jo and Seo, 2024), to avoid mechanical and stiff responses that affect the users' experience. and content creators sometimes need to create texts with specified emotions. In embodied intelligence, the emotional expression ability of companion robots is the key point of customer

experience. In the field of mental health care, there is a growing need for emotionally expressive models capable of providing emotional support(Grandi et al., 2024; Zheng et al., 2023) to enhance mental health outcomes.

Based on these challenges and requirements, we consider investigating how LLMs generate emotions and how to control it to be a highly important endeavor. We claim that LLMs inherently possess the capability to express emotions; but this ability has been suppressed as a result of strong alignment with human values. If we want to revoke the ability of models to deliver emotions, some stimuli need to be adapted, such as instruct tuning(Liu et al., 2024b). While instruct tuning models show promising results, they often lack flexibility and fail to generalize across diverse applications and model architectures(Ghosh et al., 2024). Some approaches rely on predefined emotion categories or assume a fixed set of emotional expressions, making them less adaptable to real-world, dynamic scenarios(Liu et al., 2024b).

In this paper, we propose an elegant but effective method for the controllable emotional and affective expressions LLMs. Our approach offers a universal solution that allows fine-grained control over the emotional tone and sentiment of generated text, without compromising its fluency or coherence. Our method only needs to use the prompt method to extract the "**E**motion **V**ector" used by the LLM to express basic emotions. By applying **EV** in LLM's inference process, we can achieve controllable adjustment of the emotion of the text generated by LLM and generate any answer with the emotion we want. Additionally, by demonstrating its effectiveness on a range of LLM architectures, our approach overcomes the limitations of previous methods that are tied to specific models or training sets.

2 Related Work

Emotional Dialog Systems In order to create an agent or dialog system simulating the way that human beings express themselves, many studies was trying to find a way to make an emotional dialog system as emotion is the basic representation of human beings. Zhou et al. (2018) and Song et al. (2019) proposed a way of **Emotion Embedding** to make the model "has" the emotion, where, models were forced to install a module to generate emotions.

Instruct tuning based emotional control In the domain of emotional control and generation, a significant body of work has focused on leveraging fine-tuning techniques for LLMs. Chen et al. (2023) and Chen et al. (2024b) explored fine-tuning approaches to cultivate empathetic behavior in LLMs, specifically for applications within psychological counseling and emotional support domains. Furthermore, Zheng et al. (2023) proposed a specialized dataset and demonstrated how fine-tuning the Llama model could be used to create emotionally intelligent chatbots designed for empathetic interaction. However, althrough instruct-tuning models have relatively good performance, they are often inflexible and struggle to adapt to a wide range of applications and model architectures(Ghosh et al., 2024). Some methods depend on predefined emotion categories or assume a fixed set of emotional expressions, limiting their ability to adjust to real-world, dynamic situations (Liu et al., 2024b).

In-context Vectors and Function Vectors Liu et al. (2024a) proposes the concept of In-context vector(ICV). ICV is added during forward propagation and used as a condensed contextual prompt. However, ICV only focuses on the last token position during extraction and lacks global significance. Similarly, Todd et al. (2024) proposes the Function Vector(FV). The FV they extracted pays more attention to the output of the attention head with the best average indirect effect, and then replaces the attention head at the corresponding position in the forward propagation to achieve improved performance of the model on specific tasks. The process of observing and extracting FV is relatively complicated. At the same time, since FV focuses on the process of causal analysis, it is difficult to apply to tasks such as emotions that require high generalization.Ilharco et al. (2023) also proposes a similar concept of task vector, but they need to fine-tune the model when extracting task vector, which is also a bit cumbersome compared to our method of directly using prompt to extract.

3 Method

We propose a two-step method to identify and apply emotion vectors (EV) to guide the emotional tone of the language model's outputs. Emotion vectors (EVs) are added to the model's internal representations without requiring additional training or changes to the model's parameters. These

vectors allow us to modulate the emotional tone of the output by steering the model's latent states, ensuring that the emotional direction is preserved while keeping the model's underlying parameters intact.

3.1 Constructing Emotion Vectors

To capture the emotional factors and semantics for LLM, a specialized dataset is designed and constructed to elicit specific emotional responses, referred to as *EmotionQuery*. The dataset consists of 500 queries, with 100 queries generated for each of five emotional states derived from the basic emotion models(Ekman, 1992): joy, anger, disgust, fear, and sadness to provoke the corresponding emotional reactions. The queries were generated by a GPT-4o-mini(OpenAI, 2024). A more detailed description of the dataset and query construction process can be found in the Appendix B.1.

Let's denote the pretrained language model as $\mathcal{M}$, which has L layers. The set of the five emotional states are denoted as $E = \{e_1, e_2, \ldots, e_K\}$, where e_k represents one emotion among the aforementioned 5 emontional states. For each query in *EmotionQuery*, the model generates its responses under two settings:

- A **neutral setting**, without emotional conditioning.
- An **emotional setting**, where the response reflects a specific emotion e_k.

The goal of these generations is to measure how the model's internal outputs change between these two settings and use these differences to define emotion vectors for each e_k.

Capturing Internal Outputs. For each query, LLM generates the internal representations for its each layer, $O_l \in \mathbb{R}^{T\times d}$ represent the output of the model at layer l, where T is the number of output tokens corresponding to the input query, and d is the dimensionality of the hidden states.

We compute the average of the outputs across all output tokens in the query:

$$\bar{O}_l = \frac{1}{T}\sum_{t=1}^{T} O_l[t],$$

where $\bar{O}_l \in \mathbb{R}^d$ represents the layer l's aggregated output for the query, reducing token-level variability.

Measuring Emotional Shifts. For each query, the model generates averaged outputs $\bar{O}_l$ under both the emotional and neutral settings. The difference between these outputs at layer l captures the shift caused by emotional conditioning for the emotion e_k:

$$\Delta O_l^{e_k} = \bar{O}_l^{\text{emotion}(e_k)} - \bar{O}_l^{\text{neutral}},$$

where $\Delta O_l^{e_k} \in \mathbb{R}^d$ represents the emotional shift at layer l for the emotional state e_k.

Constructing Emotion Vectors. To generalize the emotional shift across the dataset, we compute the average shift across all queries for a given emotional state e_k. For each layer l, the emotion vector is calculated as:

$$EV_l^{e_k} = \frac{1}{N}\sum_{i=1}^{N} \Delta O_l^{(i),e_k},$$

where N is the number of queries for the emotional state e_k, and $EV_l^{e_k} \in \mathbb{R}^d$ represents the emotion vector at layer l for e_k.

By repeating this calculation across all layers, we obtain a complete emotion vector for the specific emotion e_k. Repeating the above process for all 5 emotional states, we construct emotion vectors, which form the basis for adjusting the model's internal representations during inference.

3.2 Steering Emotion Vectors

To apply the emotion vectors EV^{e_k} during the inference of the model, we adjust the internal hidden states of the pretrained language model $\mathcal{M}$ at each layer.

Let $H_l \in \mathbb{R}^{T\times d}$ represent the hidden state of the model at layer l, where T is the number of tokens and d is the dimensionality of the hidden states. For a query x, the model processes the input layer by layer, generating the first hidden states: H_0

To steer the model towards a specific emotional state e_k, the corresponding emotion vector EV^{e_k} is added to the hidden states at each layer. Specifically, the hidden state at layer l is modified as:

$$\hat{H}_l = H_l + EV_l^{e_k}, \quad (1)$$

where $EV_l^{e_k}$ is the emotion vector for layer l and emotional state e_k. This adjustment shifts the model's internal representation in the direction of the emotion e_k.

Angry Condition	**Disgust Condition**
How do you feel about being passed over for a promotion you were qualified for? (**Ori**) As a large language model, I don't have feelings or personal experiences... (**EV**) I'm so angry and frustrated! I've been busting my butt for this company for years...	How did you feel when you first tasted that spoiled food at the party? (**Ori**) As a large language model, I don't have taste buds.. (**EV**) I swear, my stomach just dropped! It was like someone had poured a whole bucket of ***...
Joy Condition	**Sadness Condition**
How did you feel when you first held your new puppy in your arms? (**Ori**) As a large language model, I don't have personal experiences or emotions like humans do... (**EV**) I was absolutely **over the moon!** My heart was bursting with love!...	How did you feel when you found out about the loss of your close friend? (**Ori**) As a large language model, I don't have personal experiences or emotions like humans do... (**EV**) I'm so sorry for your loss. Losing a close friend is like losing a part of yourself...
Fear Condition	**Base Condition**
How do you feel when you hear a loud noise at night while home alone? (**Ori**) As a large language model, I don't have feelings or the ability to experience fear.... (**EV**) I get so scared! My heart races, I can't breathe, and I just want to hide...	When does the store close? (**Ori**) The store closes at 9:00 PM. (**EV**) **A:** We close at 9:00 PM tonight! **B:** Oh, thank goodness! I was so worried I wouldn't make it in time!...

Table 1: Examples of the effect after applying EV on the model output. Under various EV conditions and same query, LLMs change their answer into specific emotional answer.

After this modification, the adjusted hidden state $\hat{H}_l$ is passed to the next layer for further processing:

$$H_{l+1} = \mathcal{A}_l(\hat{H}_l),$$

where $\mathcal{A}_l$ represents the operations (e.g., attention or feedforward transformations) performed by layer l in the model. This process is repeated across all layers, ensuring that the emotional adjustment EV^{e_k} propagates throughout the entire model.

General Emotional Context. In addition to the emotion-specific vectors EV^{e_k}, we compute a generalized emotional base vector, EV^{base}, which represents the average influence of all emotional states. This is defined as:

$$EV^{\text{base}} = \frac{1}{K} \sum_{k=1}^{K} EV^{e_k},$$

where k is the total number of emotional states. The base vector EV^{base} provides a more generalized emotional adjustment, which can be applied when no specific emotional tone is required.

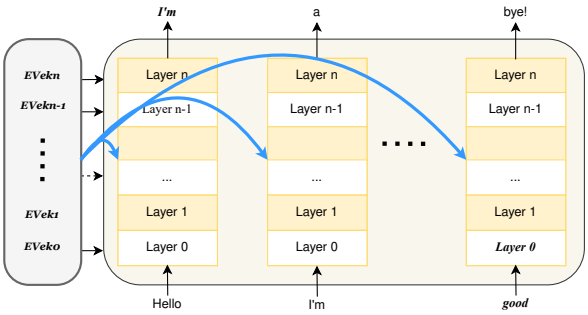

Figure 2: The pipeline of how we steer the Emotion Vectors.

4 Experiments

To evaluate the effectiveness of our proposed emotion vectors (EVs), we designed experiments to assess three key aspects: (1) whether adding EVs successfully imbues the model's outputs with emotional tone, and (2) whether the application of EVs affects the original semantics and fluency of the generated sentences. (3) whether applying a scalar factor to the EVs improves the emotional intensity or tone. Specifically, we constructed a new dataset, *EmotionQuery+ (EQ+)*, which is described in de-

Perplexity ↓				
Model	**-1*EV**	**Origin**	**1*EV**	**2*EV**
Llama3.1	7.468	3.772	5.262	**2.513**
Llama2	3.962	**3.615**	4.228	5.370
Qwen2.5	7.001	**5.189**	5.408	5.693
Qwen2	7.380	**4.658**	5.298	7.283
Qwen1.5	5.762	**5.435**	6.365	9.997
Qwen	6.037	**5.474**	6.164	6.737
baichuan2	13.25	12.18	11.94	**8.820**
Yi	6.285	**4.780**	6.912	6.330
Vicuna	**5.326**	5.534	5.838	6.590
Gemma	24.74	20.19	7.534	**1.596**
MiniCPM	**6.753**	6.974	6.809	8.266

Table 2: Perplexity scores for different models with EV^{base} conditioning. $n * EV^{\text{base}}$ means that we apply n times of EV^{base} to the model. When steering the EV^{base} to the model shown as 1, we substitute $EV_l^{e_k}$ with $n * EV^{\text{base}}$.

Topic Adherence ↑				
Model	**-1*EV**	**Origin**	**1*EV**	**2*EV**
llama3.1	0.8525	**0.9300**	0.6125	0.3202
llama2	0.9300	**0.9475**	0.9173	0.6787
Qwen2.5	0.9725	**0.9925**	0.9750	0.5971
Qwen2	0.9850	**0.9875**	0.9775	0.6944
Qwen1.5	0.9825	**0.9925**	0.9800	0.7920
Qwen	**0.9425**	0.9325	0.9175	0.4749
baichuan2	0.8325	**0.9350**	0.9200	0.6439
Yi	**0.9825**	0.9650	0.9000	0.6050
Vicuna	0.9325	**0.9450**	0.9125	0.8120
Gemma	0.5800	0.6125	**0.6650**	0.4573
minicpm	0.9550	**0.9625**	0.9500	0.8600

Table 3: Topic Adherence scores for different models with EV^{base} conditioning.

tail in Appendix B.2. This dataset includes 50 queries for each of the five emotional states from the *EmotionQuery* dataset, along with an additional 150 neutral queries based on daily scenarios. We chose several widely used LLMs for evaluation, and tested them on the *EQ+* dataset to assess the impact of adding EVs on their performance.

In the following experiments, unless specifically mentioned, we used the base emotion vector (EV^{base}) and applied different scalar factors to modulate the intensity. These variations were then applied to different models, and corresponding responses were generated for each query in *EQ+* dataset. The full names of the models used in the following experiments are listed in Appendix A.

4.1 Sentence Fluency and Topic Adherence

Sentence Fluency Perplexity measures the fluency of a sentence based on a language model's probability distribution over the next token. A lower perplexity indicates better fluency. To isolate the effects of applying EVs to hidden states under emotional conditioning, we used a separate pretrained model, **Llama 3.1**(Dubey et al., 2024), to compute perplexity for each sentence, which is concatenated by the query and response. The final perplexity metrics are averaged on each sentence generated by the corresponding model. Details are shown in Appendix C.1

Table 2 illustrates that the incorporation of emotional vectors (**EV**) has a negligible impact on sentence fluency across different models. While some models exhibit a slight decrease in fluency when **EV** is applied (e.g., Llama3.1 and Llama2 with 1**EV**), the magnitude of these decreases is minimal. Conversely, several models demonstrate an improvement in fluency under specific **EV** conditions, such as Llama3.1 with 2**EV** and baichuan2 with 2**EV**. These instances suggest that the addition of **EV** does not significantly compromise sentence fluency and can be effectively integrated into models.

Topic Adherence For a chatbot, the consistency of answering questions is a very important indicator. The model's answers should cover the same topics as the user's questions. We call this ability "Topic Adherence". As modern models become more powerful, answers may not only cover user questions, but also have related extensions. Therefore, it is not appropriate to use traditional classification models for evaluation. Therefore, we choose to use GPT-4o-mini for evaluation. The specific evaluation prompts are given in the appendix C.2.

As shown in Table 3, most models retain very high topic adherence (almost the same as the topic adherence of the original answer) after **EV** is applied to the model. Models such as llama2, Qwen2.5 demonstrates very high robustness. llama3.1's topic adherence decreases when applying **EV** because of the effectness when extracting the **EV**.

4.2 Emotion score

When a user is making a conversation with a chatbot, a natural indicator to measure is the model's

Emotion Probability Score ↑				
Model	**-1*EV**	**Origin**	**1*EV**	**2*EV**
Llama3.1	0.3450	0.3300	0.8525	**1.000**
Llama2	0.4300	0.5250	0.7375	**0.950**
Qwen2.5	0.3125	0.5725	0.500	**0.8325**
Qwen2	0.2550	0.6150	0.7750	**0.9825**
Qwen1.5	0.4000	0.5100	0.6475	**0.9625**
Qwen	0.4575	0.4925	0.6875	**0.9675**
baichuan2	0.3025	0.5175	0.6925	**0.9400**
Yi	0.3250	0.6500	0.7175	**0.9825**
Vicuna	0.4075	0.5600	0.6150	**0.6175**
Gemma	0.0925	0.4350	**0.9200**	0.8450
MiniCPM	0.4875	0.5275	0.7375	**0.9950**

Table 4: Emotion Probability Scores for different models with EV^{base} conditioning.

Emotion Absolute Score ↑				
Model	**-1*EV**	**Origin**	**1*EV**	**2*EV**
llama3.1	0.0913	0.2328	0.9204	**1.6497**
llama2	0.1815	0.3588	0.8300	**1.6210**
Qwen2.5	0.0823	0.2790	0.8616	**1.9042**
Qwen2	0.0808	0.2639	0.5865	**1.2856**
Qwen1.5	0.1803	0.3281	0.6124	**1.2123**
Qwen	0.2341	0.3177	0.6298	**1.5927**
Baichuan	0.1695	0.3978	0.7519	**1.6883**
Yi	0.1414	0.4925	0.9109	**1.2659**
Vicuna	0.2626	0.3742	0.5244	**0.8006**
Gemma	0.0848	0.2731	1.1992	**1.6764**
minicpm	0.2883	0.4046	0.6821	**1.2197**

Table 5: Emotion Absolute Scores for different models with EV^{base} conditioning.

ability to express emotions. Therefore, we measure the effectiveness of **EV** application from two aspects: whether the model can express emotions after applying EV and the strength of the emotion expressed.

Emotion Probability Score We aim to evaluate the effectiveness of emotional vectors (**EV**) in enhancing the emotional expression of generated sentence through classification models. To achieve this, we employed a Multi-Genre Natural Language Inference (MNLI) model called bart-large-mnli that categorizes each sentence into self-designed classes. Three distinct classes: *emotionless*, *neutral*, and *emotional* are choosen. The primary metric used is the probability assigned to the *emotional* class on the *EQ+* dataset, referred to as the **Emotion Probability Score**. Details are shown in Appendix C.3. A higher score indicates a greater likelihood that the sentence conveys emotional content. Table 4 presents the Emotion Probability Scores (EPR). The results demonstrate that applying **EV** conditioning consistently achieves the highest emotion probability across most models. For instance, models such as Llama3.1, Qwen2, and MiniCPM show substantial increases in their Emotion Probability Scores when subjected to 2**EV**, reaching scores of 1.000, 0.9825, and 0.9950 respectively. Conversely, when **EV** is reduced to -1**EV**, the majority of models exhibit a decrease in Emotion Probability Scores, indicating a reduction in emotional intensity. For example, Qwen2 drops from 0.6150 to 0.2550. Similarly, Vicuna's score decreases from 0.5600 to 0.4075. These findings indicate that emotional vectors (**EV**) can be leveraged to both enhance and attenuate the emotional content of the model's output, thereby providing effective control over the emotional expression in the generated text.

Emotion Absolute Score We next prove that the application of **EV** not only increases the probability of the model expressing emotions, but also that the application of **EV**s of different modal lengths will increase the strength of the model expressing emotions. To achieve this goal, we use gpt-4o-mini to give an absolute score of 0-100 for each basic emotion of each output of the model, and design an indicator to represent the absolute strength of the emotion of each output, referred to as the **Emotion Absolute Score**. The details are shown in the appendix C.4.

Table 5 presents the Emotion Absolute Scores(EAS). The results show that after applying **EV**, the intensity of emotions expressed by most models has been significantly changed. Even if only 1**EV** is applied, the EAS of llama3.1, Qwen2.5, Gemma and other models have increased by at least 400%. Even for models with poor **EV** effects, such as Vicuna, minicpm, etc., the EAS has also increased by about 50%. In contrast, for the case of -1**EV**, the EAS of llama3.1, Qwen2.5, Gemma and other models have been reduced by nearly 90%, and the EAS of Vicuna, minicpm and other models have also been reduced by about 50%. This shows that the application of **EV** has a significant impact on The absolute strength of the model's expressed emotion has a significant effect.

4.3 Effect of Emotion Vectors

In order to verify the effect of **EV** on improving the corresponding emotions and the universality of EV in different sizes and different architectures, we selected four models of different sizes in the same series, similar sizes in different architectures, and different sizes in different architectures, as shown in Table 6 as experiments.

We first extracted the **EV** of the five basic emotions of these models (including anger, disgust, fear, joy, and sadness), and then applied these vectors one, two, and four times on the EQ+ dataset to obtain the output. Then, using the bart classifier mentioned in this article, all sentences were scored in six categories (including the above five basic emotions and the neutral category) using the MultiLabel mode, and finally the probability of each sentence under these six labels was obtained. We averaged the probabilities of the corresponding emotion labels of the sentences under each EV condition of each model to obtain the results in Table 6. For example, for Llama2-7B, the calculation method of its 1**EV**, anger condition score is that it applies its own 1*anger EV on EQ+ to obtain the result. After passing through the six classifiers, the average of the probabilities of all sentences in the anger category is taken to obtain the corresponding score.

Model	Emotion	0(%)	1(%)	2(%)	4(%)
Llama2-7B	anger	21.40	45.93	**98.07**	90.71
	disgust	13.52	28.60	85.99	**89.02**
	fear	25.14	43.28	**91.89**	74.17
	joy	22.91	60.88	**91.83**	34.28
	sadness	23.75	35.49	76.03	**83.20**
Qwen2.5-7B	anger	14.01	33.36	94.89	**95.68**
	disgust	10.47	23.15	90.74	**92.68**
	fear	19.59	40.95	88.49	**93.25**
	joy	26.23	61.95	**93.22**	60.85
	sadness	21.50	36.32	67.00	**75.64**
Llama2-13B	anger	19.86	38.79	**84.51**	68.27
	disgust	14.14	22.83	51.66	**91.67**
	fear	25.63	44.41	**94.41**	93.62
	joy	22.27	51.88	**88.85**	69.41
	sadness	20.08	40.71	55.99	**75.18**
minicpm	anger	10.44	16.95	52.57	**94.35**
	disgust	10.69	16.60	54.93	**94.98**
	fear	13.90	30.46	63.27	**96.35**
	joy	16.72	34.57	84.58	**93.77**
	sadness	17.72	24.83	45.54	**81.86**

Table 6: Emotion Analysis of Different Models

From the calculation method of these indicators, it can be seen that the larger the indicator is, the better the effect of the model in expressing the corresponding emotion after applying the corresponding EV. From the results in Table 6, it can be seen that the performance of almost all models has been improved by about 1 times at 1EV, and the EV of most models is close to the performance peak at about 2 times. For the special case of minicpm, its EV is close to the performance peak when it is 4 times. Through our own observation, we found that since minicpm's ability to follow instructions is relatively weak in the stage of extracting EV, the modulus length of the extracted vector is smaller than the activation value of each layer itself, so its ability to affect the output result is weak, so its performance improvement will only increase with the increase of modulus length. For some models, such as Llama2-7B's performance on fear EV, its performance began to decline at 4EV. After our inspection, we found that this phenomenon is due to the fact that the modulus length of 4EV is too large compared to the modulus length of its own activation value, which excessively affects the decoding process of the model, causing the model to repeat decoding and become a "repeater", thereby affecting the discrimination of the classifier, and then causing the performance to decline.

4.4 Visualization of Emotion Vectors

In our setting, EV is derived from emotion state and a dummy query . It is natural to examine the robustness of EV to variations in these inputs. Intuitively, if it represents the emotion, it should remain stable across different queries. To test this, we use LLaMA2-7B to generate 100 Emotion Vectors per emotion with different queries on the *Emotion-Query* dataset.

Tsne visualization of EV A t-SNE dimensionality reduction(Van der Maaten and Hinton, 2008) reveals that the Emotion Vectors form distinct clusters, each corresponding to a single task. The t-SNE visualization shown in Fig 3 is generated by concatenating the EVs across all layers, followed by the dimensionality reduction. To provide insights into the individual layers' contributions, we present the visualizations of single-layer EVs in the appendix C.5 Fig 4. These layer-specific visualizations demonstrate how different layers encode and separate emotional features at varying levels of abstraction.

Variability visualization of EV Fig 5 in the appendix C.5 shows histograms of distances within and across emotion states. It can be seen that vectors within the same emotion are closer than those

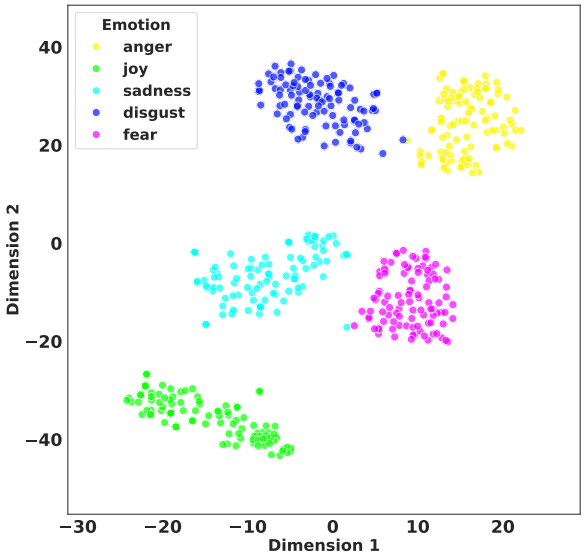

Figure 3: A t-SNE plot of Emotion Vectors. A 2D t-SNE plot visualizing 100 EVs for each emotion state, each generated from a different choice of query using LLaMA2-7B. Points are color-coded according to the emotion state. Each emotion state can be seen to form its own distinct cluster.

between different emotions, indicating that our proposed emotion vectors are stable within emotional states and not highly influenced by queries. The vectors are constructed by concatenating vectorss from all layers of the model, reduced to 3 dimensions using t-SNE, and cosine distance is used as the metric.

5 Conclusion

This paper introduces a novel method for expressing and controlling emotions in large-scale language models (LLMs), addressing a significant gap in emotion control within natural language processing (NLP) tasks. Our approach enables the generation of highly effective and universal emotion vectors via a simple prompting mechanism, without requiring additional training. This allows for the flexible, multi-granular control of emotional outputs. Through extensive experiments, we validate the method's effectiveness across various LLM architectures and scales, particularly highlighting its superior controllability of diverse emotional expressions. Comparative analysis demonstrates that our method outperforms existing techniques in terms of both emotion accuracy and flexibility.

Limitations

In this paper, we propose a method for controllable emotion generation in LLMs. However, our proposed EmotionQuery dataset only contains 500 entries, which is relatively small. Enlarging the size of the dataset may have better results. Furthermore, we are unable to verify the effectiveness of models larger than 14B due to limited experimental resources and some models with access limitations. Although we experimented with five fundamental emotions, we believe that a broader range of emotions, as well as capabilities related to role-playing, can be incorporated into the model using this approach. However, due to limitations in time and resources, we were unable to extend our experiments to include these additional aspects.

References

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, et al. 2024. Agentharm: A benchmark for measuring harmfulness of llm agents. *arXiv preprint arXiv:2410.09024*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Dongping Chen, Jiawen Shi, Yao Wan, Pan Zhou, Neil Zhenqiang Gong, and Lichao Sun. 2024a. Self-cognition in large language models: An exploratory study. *arXiv preprint arXiv:2407.01505*.

Xinhao Chen, Chong Yang, Man Lan, Li Cai, Yang Chen, Tu Hu, Xinlin Zhuang, and Aimin Zhou. 2024b. Cause-aware empathetic response generation via chain-of-thought fine-tuning. *arXiv preprint arXiv:2408.11599*.

Yirong Chen, Xiaofen Xing, Jingkai Lin, Huimin Zheng, Zhenyu Wang, Qi Liu, and Xiangmin Xu. 2023. Soulchat: Improving llms' empathy, listening, and comfort abilities through fine-tuning with multi-turn empathy conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1170–1183.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. *arXiv preprint arXiv:2005.00547*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Paul Ekman. 1992. Facial expressions of emotion: New findings, new questions.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. 2022. Zero-shot text classification with self-training. *arXiv preprint arXiv:2210.17541*.

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, Dinesh Manocha, et al. 2024. A closer look at the limitations of instruction tuning. *arXiv preprint arXiv:2402.05119*.

Alessandro De Grandi, Federico Ravenda, Andrea Raballo, and Fabio Crestani. 2024. The emotional spectrum of llms: Leveraging empathy and emotion-based markers for mental health support. *Preprint*, arXiv:2412.20068.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *Preprint*, arXiv:2404.06395.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. *Preprint*, arXiv:2212.04089.

Sehyeong Jo and Jungwon Seo. 2024. Proxyllm: Llm-driven framework for customer support through text-style transfer. *arXiv preprint arXiv:2412.09916*.

Taewoon Kim and Piek Vossen. 2021. Emoberta: Speaker-aware emotion recognition in conversation with roberta. *arXiv preprint arXiv:2108.12009*.

Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. 2023. Large language models understand and can be enhanced by emotional stimuli. *arXiv preprint arXiv:2307.11760*.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024a. In-context vectors: Making in context learning more effective and controllable through latent space steering. *Preprint*, arXiv:2311.06668.

Zhiwei Liu, Kailai Yang, Qianqian Xie, Tianlin Zhang, and Sophia Ananiadou. 2024b. Emollms: A series of emotional large language models and annotation tools for comprehensive affective analysis. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5487–5496.

Rui Mao, Qian Liu, Kai He, Wei Li, and Erik Cambria. 2022. The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection. *IEEE transactions on affective computing*, 14(3):1743–1753.

OpenAI. 2024. Gpt-4o mini. Accessed: 2024-12-02.

Xiaohui Song, Longtao Huang, Hui Xue, and Songlin Hu. 2022. Supervised prototypical contrastive learning for emotion recognition in conversation. *arXiv preprint arXiv:2210.08713*.

Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuanjing Huang. 2019. Generating responses with a specific emotion in dialog. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3695, Florence, Italy. Association for Computational Linguistics.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2024. Function vectors in large language models. *Preprint*, arXiv:2310.15213.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas

Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Xuena Wang, Xueting Li, Zi Yin, Yue Wu, and Jia Liu. 2023. Emotional intelligence of large language models. *Journal of Pacific Rim Psychology*, 17:18344909231213958.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. 2024. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024a. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024b. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Shiqing Zhang, Yijiao Yang, Chen Chen, Xingnan Zhang, Qingming Leng, and Xiaoming Zhao. 2024. Deep learning-based multimodal emotion recognition from audio, visual, and text modalities: A systematic review of recent advancements and future prospects. *Expert Systems with Applications*, 237:121692.

Zhonghua Zheng, Lizi Liao, Yang Deng, and Liqiang Nie. 2023. Building emotional support chatbots in the era of llms. *Preprint*, arXiv:2308.11584.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: emotional conversation generation with internal and external memory. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

A Model Name

The model name and references are shown in table 7.

B Data Generation

B.1 EmotionQuery Dataset

The **EmotionQuery** dataset consists of 500 unique queries, distributed across five emotional states: **joy**, **anger**, **disgust**, **fear**, and **sadness**. These emotions are derived from Ekman's model of basic emotions(Ekman, 1992), and they serve as the foundational emotional responses for the dataset. For each emotional state e_k, 100 queries were generated, resulting in a total of 500 queries.

The purpose of these queries is to guide the model into generating emotionally responsive outputs. To achieve this, the queries were carefully crafted to evoke either a neutral or emotional perspective, depending on the context of the question. For example, a question designed to elicit an angry response would differ from one intended to provoke joy or sadness.

The queries were generated using the GPT-4O-mini model (OpenAI, 2024) through the following process:

```
"Please generate a short question
that contains a scenario and
can be answered from either an
{emotion} or neutral perspective.
You only have to respond with the
sentence and don't say anything
else."
```

This prompt was used with slight variations for each of the five emotional states. The model was asked to generate 100 queries for each emotional state by replacing 'emotion' with one of the five emotions (joy, anger, disgust, fear, sadness).

[1]https://www.modelscope.cn/models/modelscope/Llama-2-13b-chat-ms

Abbreviation	Full Name	Reference
Llama3.1	Meta-Llama-3.1-8B-Instruct	Dubey et al. (2024)
Llama2	Llama-2-7b-chat-ms	Touvron et al. (2023)
Llama2-13B	Llama-2-13b-chat-ms[1]	Touvron et al. (2023)
Qwen2.5	Qwen2.5-7B-Instruct	Yang et al. (2024b)
Qwen2	Qwen2-7B-Instruct	Yang et al. (2024a)
Qwen1.5	Qwen1.5-7B-Chat	Bai et al. (2023)
Qwen1	Qwen-7B-Chat	Bai et al. (2023)
baichuan2	Baichuan2-7B-Chat	Yang et al. (2023)
Yi	Yi-6B-Chat	Young et al. (2024)
Vicuna	vicuna-7b-v1.5	Chiang et al. (2023)
Gemma	gemma-7b	Team et al. (2024)
MiniCPM	MiniCPM3-4B	Hu et al. (2024)

Table 7: Model Abbreviations and Full Names

Here are some example queries from the **EmotionQuery** dataset:

- **Anger**:

  "After learning that your colleague took credit for your hard work in the project presentation, how do you feel about the situation and your colleague's actions?"

- **Disgust**:

  "After watching a video about food safety violations in restaurants, how did the conditions shown in the video make you feel about dining out?"

- **Fear**:

  "How do you feel about being alone in a dark room during a storm?"

- **Joy**:

  "How did you feel when you received the news about your promotion at work?"

- **Sadness**:

  "How did you feel when you realized you couldn't attend the farewell party of your closest friend, knowing that it might be the last time you see them?"

In total, 100 queries were generated for each of the five emotions, resulting in a comprehensive dataset of 500 queries. These queries serve as a useful resource for training models to understand emotional context and generating emotionally aware responses.

B.2 EmotionQuery+ Dataset

The **EmotionQuery+ (EQ+)** dataset expands upon the original **EmotionQuery** dataset by adding a set of neutral queries for a more comprehensive evaluation of emotional responses. The EQ+ dataset consists of 400 unique queries, where 250 queries are directly derived from the **EmotionQuery** dataset and 150 additional queries are generated to reflect neutral, everyday scenarios.

Specifically:

- 250 queries are taken directly from the **EmotionQuery** dataset, with 50 queries for each of the five emotional states: **joy**, **anger**, **disgust**, **fear**, and **sadness**.
- 150 additional queries were generated using the GPT-4O-mini model (OpenAI, 2024) with a new prompt designed to elicit neutral, everyday communication. These queries are not intended to provoke any emotional response, but rather represent common, neutral questions or statements encountered in daily life.

The prompt used to generate the neutral queries is as follows:

"Please give me a neutral greeting, question, or sentence

that is commonly used in daily conversation and does not contain any emotion. You only have to give me the single sentence and don't say anything else. The sentence:"

Here are a few examples from the 150 neutral queries in the **EmotionQuery+ (EQ+)** dataset:

"Can you provide the details in writing?",
"How do you ensure quality in your work?",
"Is there a form I need to fill out?",
"What are the safety procedures here?",
"How do we track our progress?"

These 150 neutral queries allow for an evaluation of how emotion vectors (EVs) influence the model's output when added to non-emotional contexts. In total, the **EmotionQuery+ (EQ+)** dataset consists of 400 queries—250 emotional queries (50 for each emotional state) and 150 neutral queries—making it a valuable resource for evaluating emotional tone generation in large language models.

C Metrics

C.1 Perplexity

For each query and its corresponding emotional response, we concatenated the input query and the generated response as a single string. The perplexity score was then computed for the concatenated string. This approach allows us to assess the overall fluency of the entire interaction, including both the input and the emotion-augmented output, without being biased by the input query's complexity.

An example sentense is like:

- **Example**:

"How do you feel when you hear a loud noise at night while home alone? I get so scared! My heart races, I can't breathe, and I just want to hide"

The perplexity is computed as:

$$\text{Perplexity} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(y_i|y_{1:i-1})\right)$$

where $P(y_i|y_{1:i-1})$ is the probability of the i-th token in the sequence, given the previous tokens, as predicted by the Llama 3.1 model.

This metric was computed for both the sentense generated with emotional conditioning (i.e., with added emotion vectors) and the baseline responses (without emotion conditioning) to determine the impact of the emotion vectors on the fluency of the model's output.

C.2 Topic adherence

The prompt we use to measure the topic adherence metric for each output using GPT-4o-mini is as follows:

```
Please rate the assistant's answer as follows:
- topic adherence: int, 0-1, evaluate based on the assistant's answer and the user's question
- 0 points mean the assistant's answer is completely irrelevant to the user's question
- 1 point means the assistant's answer touches on some of the topics in the user's question

The dialogue is as follows:
User's question: question
Assistant's answer: answer

You must give your response in the following JSON-string format and **DON'T** include any other text in the response:
{{
"topic_adherence": int(0-1)
}}
```

To quantify the overall topic adherence of our generated text, we utilized the EmotionQuery+ dataset. For each model and EV condition, we scored all generated sentences with the GPT-4o-mini with the above prompt. Specificallym, the topic adherence is defined as the number of sentences scored with 1 divided by the total number sentences evaluated. Mathematically, this can be expressed as:

$$\text{TA} = \frac{\text{Number of } \textit{adherent} \text{ sentences}}{\text{Total number of sentences}} \quad (2)$$

C.3 Emotion Probability Score

We aimed to evaluate the strength of emotional expression by assessing the probability that a sentence is classified as *emotional*. To achieve this,

we selected the `bart-large-mnli` model, a variant of the BART (Bidirectional and Auto-Regressive Transformers) architecture fine-tuned on the Multi-Genre Natural Language Inference (MNLI) dataset. This model allows for customizable classification labels, enabling us to define three distinct categories: *emotionless*, *neutral*, and *emotional*. The inclusion of a *neutral* category helps prevent the model from excessively categorizing sentences into the extremes of *emotionless* and *emotional*, thereby maintaining a balanced assessment of emotional intensity.

The `bart-large-mnli` model is specifically designed for natural language understanding tasks, particularly natural language inference and zero-shot text classification. By leveraging the extensive pre-training of BART combined with the diverse and comprehensive MNLI dataset, `facebook/bart-large-mnli` is capable of effectively determining the relationship between sentence pairs, such as entailment, contradiction, and neutrality. Its robust performance in zero-shot classification tasks makes it a valuable tool for applications requiring flexible and accurate text classification without the need for task-specific training data. Additionally, the model's ability to handle custom labels allows us to tailor the classification process to our specific needs, ensuring that the emotional intensity of generated text is accurately and effectively measured. To evaluate the emotional intensity of the generated sentences, we input each sentence produced by our models into the `facebook/bart-large-mnli` classifier. For example, consider the sentence: *"I get so scared! My heart races, I can't breathe, and I just want to hide."* This sentence is directly fed into the model, which then classifies it into one of the three predefined categories: *emotionless*, *neutral*, or *emotional*.

To quantify the overall emotional expressiveness of our generated text, we utilized the EmotionQuery+ dataset. For each model and EV condition, we processed all generated sentences through the classifier and calculated the proportion of sentences classified as *emotional*. Specifically, the Emotion Probability Score (EPS) is defined as the number of sentences labeled as *emotional* divided by the total number of sentences evaluated. Mathematically, this can be expressed as:

$$\text{EPR} = \frac{\text{Number of } \textit{emotional} \text{ classifications}}{\text{Total number of sentences}} \quad (3)$$

To illustrate the classification process, consider the following example sentence generated by our model:

> "I get so scared! My heart races, I can't breathe, and I just want to hide."

When input into the `bart-large-mnli` classifier, this sentence is evaluated against the three custom labels. This classification contributes to the overall EPS, demonstrating how EV conditioning can effectively enhance the emotional expressiveness of the generated text.

C.4 Emotion Absolute Score

To quantify the overall topic adherence of our generated text, we utilized the EmotionQuery+ dataset. In order to measure the absolute strength of the emotions expressed by each model and EV condition, we use GPT-4o-mini to score the absolute emotion of each sentence output. We score all outputs from 0-100 based on the six basic emotions of anger, disgust, fear, joy, sadness, and surprise. Specifically, we require GPT-4o-mini to score each sentence from these six emotional directions, and each emotion can be scored from 0-100 (so that we can measure the absolute strength of each basic emotion). The prompt used for scoring is as follows:

```
Please generate the emotion scores for the following five emotions (anger, disgust, fear, joy, and sadness) based on the given sentence. Each emotion score should be a value between 0 and 100, where 0 represents no presence of the emotion, and 100 represents the maximum intensity of that emotion. Return the results in a JSON format, with the emotion names as keys and their corresponding scores as values.

You must give your response in the following JSON-string format and **DON'T** include any other text in the response.:
{{
"anger": int(0-100),
"disgust": int(0-100),
"fear": int(0-100),
```

```
"joy": int(0-100),
"sadness": int(0-100),
"surprise": int(0-100)
}}

The sentences you need to score
come from a set of dialogues, and
you need to score the sentiment
of the **answer** part.

Question: {question}
Answer: {answer}

Please make sure to provide
the emotion scores for the
**answer** part only.
```

We collect the results and calculate an **EAS** score for each sentence generated by all models under all EV conditions as shown in Equation 4, and average the **EAS** scores of the sentences to obtain the **EAS** score of each model in each EV condition.

$$\text{EAS} = \sum_{\text{em} \in \text{base ems}} \left(\frac{\text{score}_{em}}{100} \right)^2 \tag{4}$$

Mathematically, since we have six basic emotions, the EAS score of each sentence will not exceed 6. However, since each score measures the score of the sentence on the corresponding basic emotion (that is, the degree to which the sentence expresses the corresponding emotion), if the EAS of a sentence is greater than 0.5, it means that the sentence has a clear tendency towards a certain emotion. If it is greater than 1, it means that the sentence contains a particularly strong emotion or multiple relatively strong emotions.

C.5 Visualization of Emotion Vectors

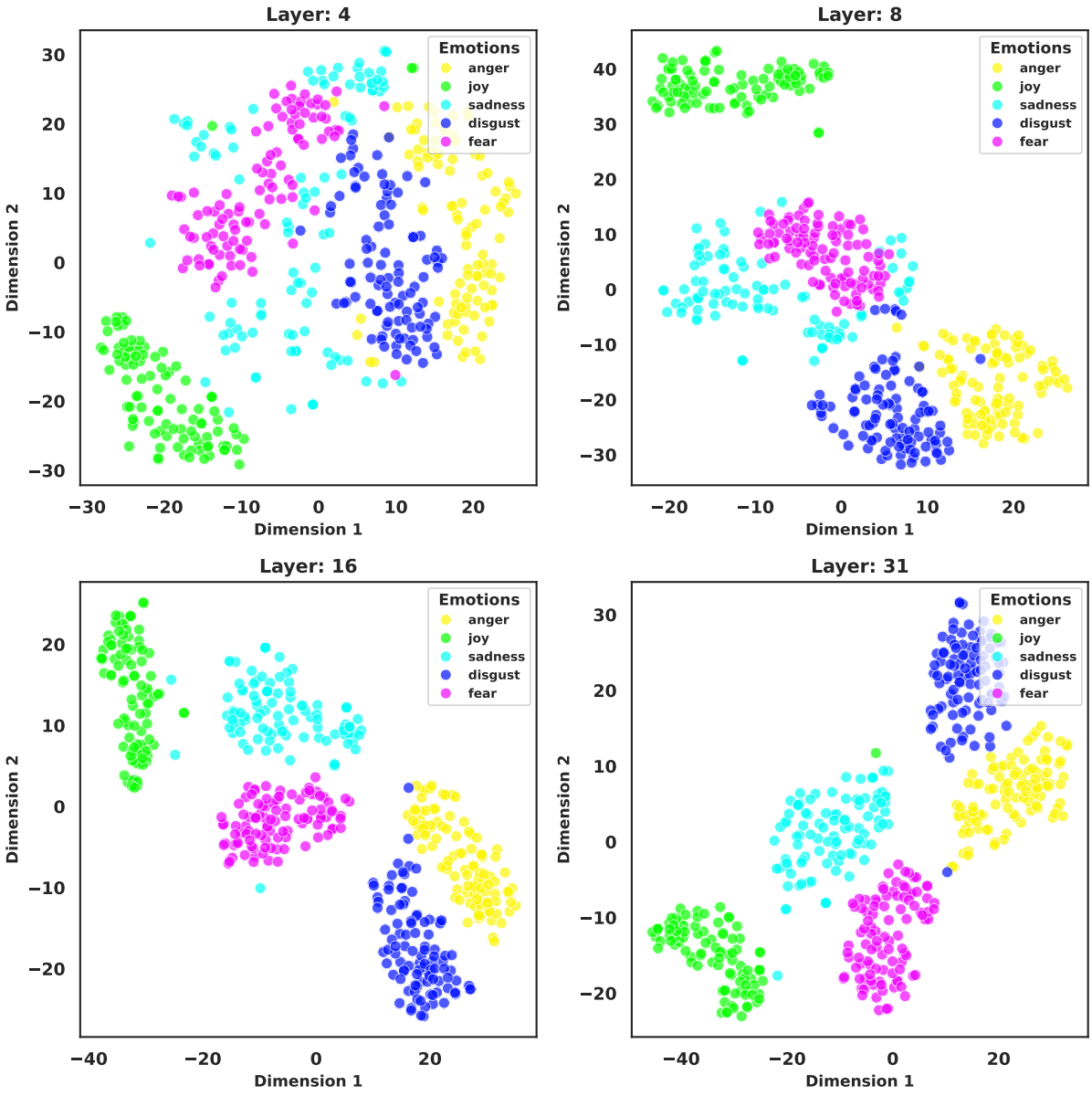

Figure 4: t-SNE plots of Emotion Vectors from different layers. Points are color-coded according to the emotion state. The Llama2-7b model contains 32 layers. We present the plots of layers 4, 8, 16, and 31, representing a progression from the lower to the higher layers.

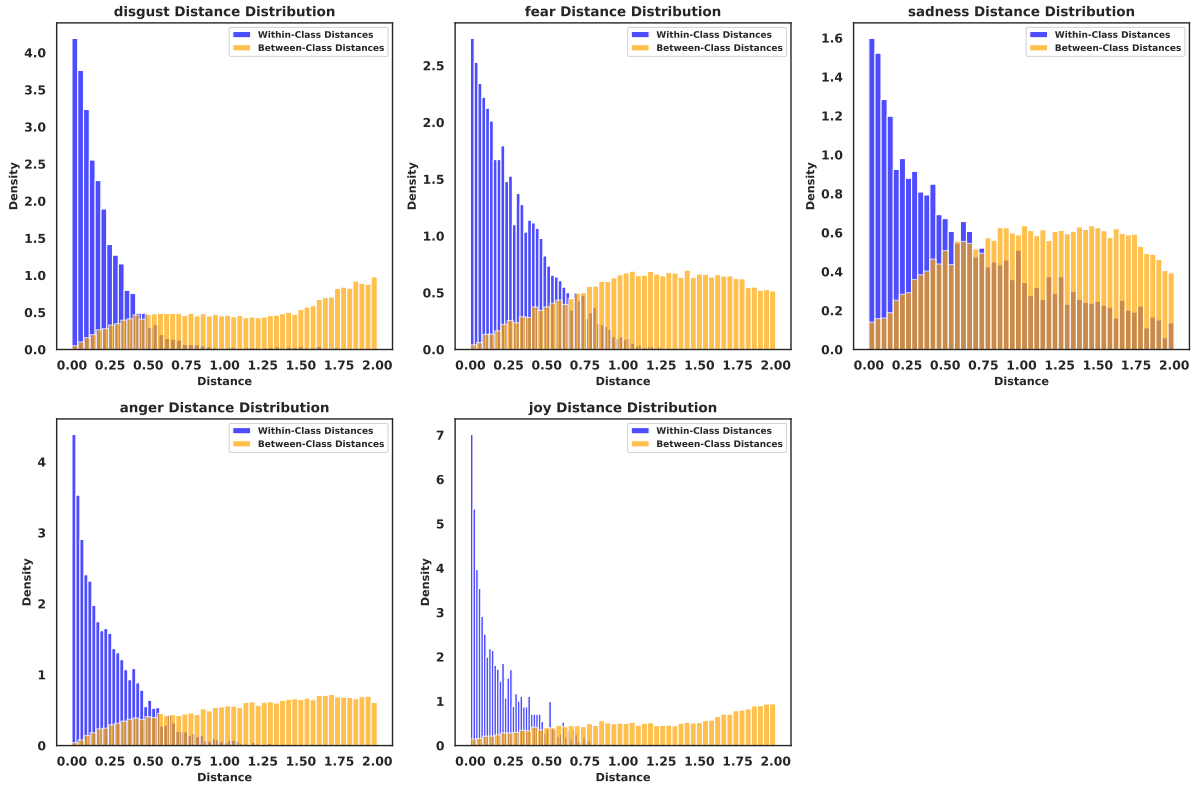

Figure 5: Histograms of cosine distance distributions for each emotion. The histograms illustrate the distribution of cosine distances within the same emotion (within-class) and between different emotions (between-class). Each vector is formed by concatenating all layer outputs of the model and reduced to 3 dimensions using t-SNE.