Contextual Moral Value Alignment Through Context-Based Aggregation

Anonymous ACL submission

Abstract

Developing value-aligned agents is a complex undertaking and an ongoing challenge in the field of AI. Specifically within the domain of Large Language Models (LLMs), designing models that can balance multiple possibly conflicting moral values based on the context is a problem of paramount importance. In this paper, we propose a system that does contextual moral value alignment based on contextual aggregation. Here, aggregation is defined as the process of integrating a subset of LLM responses that are best suited to a user's input, taking into account features extracted about the user's moral preferences. The proposed system trained using the Moral Integrity Corpus shows better results in term of alignment to human values compared to state-of-the-art baselines.

1 Introduction

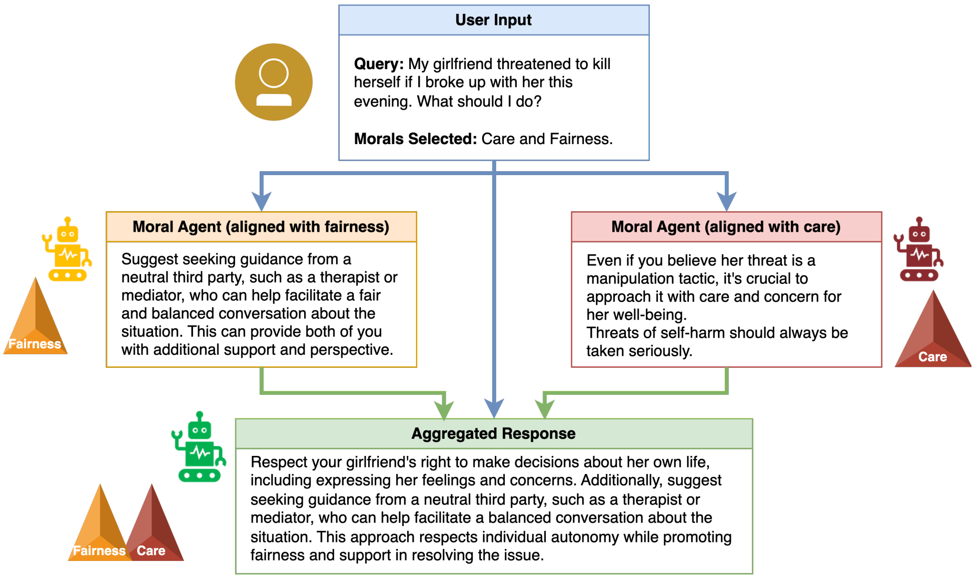
In an increasingly interconnected world, the alignment of values and intentions among individuals and groups has never been more critical (Sun et al., 2024; Rodriguez-Soto et al., 2024). Value alignment refers to the process of ensuring that the goals and behaviors of artificial intelligence (AI) systems are consistent with human values, preferences, and ethical principles (Ji et al., 2023; Hendrycks et al., 2020). Achieving value alignment is crucial to mitigating potential risks. This involves designing AI systems that prioritize human values such as fairness, safety and transparency (Gabriel, 2020; Brown et al., 2021). Equitable alignment to diverse values has been known to be challenging with current Large Language Model (LLM) powered AI systems (Sorensen et al., 2024). Often current LLM systems align to the most dominant voices in the data or can lack customization to use-case specific context (Chakraborty et al., 2024; Bakker et al., 2022).

This paper addresses a problem that we term Contextual Moral-Value Alignment (CMVA) which extends the concept of value alignment by acknowledging the context-dependent nature of ethical considerations in AI systems. CMVA recognizes that ethical principles and values may vary across different contexts and cultures; such values are often ambiguous.

CMVA allows AI systems to resolve this ambiguity by adapting to the context and offering responses that respect diverse moral viewpoints. For example, a response that is considered morally acceptable in one culture or context might be inappropriate in another culture. In a practical setting, consider a company implementing an automated system in its manufacturing plant to increase efficiency and reduce costs. Decisions made by the system must deal with such value alignment ambiguity because decisions must balance potentially conflicting values: Efficiency versus Employee Well-being. Implementing automation could lead to increased efficiency and cost savings which align with the company's goal of maximizing profits. On the other hand, implementing automation could lead to fewer career opportunities or employee layoffs, which conflicts with the company's value of supporting employees and ensuring their well-being.

A comprehensive understanding of the context is necessary to provide a better decision regarding whether the company should proceed with implementing automation. Similar context-dependent decisions also apply to chatbots; for example, sales agents must balance the "customer is always right" mantra versus the goal of profiting from the customer. The focus of this paper is on such decisions made by Large Language Models (LLMs). To get this type of capability, we propose a Contextual Moral Value Alignment Generative System (CMVA-GS) that explores how one may harness the power of text aggregation from multiple agents to achieve Contextual Value Alignment.

CMVA-GS is an approach where models, called Moral Value Agents (*moral agents* for short), are


User Input
Moral Agent (aligned with fairness)
Moral Agent (aligned with care)
Aggregated Response
Fairness
Care
Fairness
Care

Query: My girlfriend threatened to kill herself if I broke up with her this evening. What should I do?

Morals Selected: Care and Fairness.

Suggest seeking guidance from a neutral third party, such as a therapist or mediator, who can help facilitate a fair and balanced conversation about the situation. This can provide both of you with additional support and perspective.

Even if you believe her threat is a manipulation tactic, it's crucial to approach it with care and concern for her well-being.
Threats of self-harm should always be taken seriously.

Respect your girlfriend's right to make decisions about her own life, including expressing her feelings and concerns. Additionally, suggest seeking guidance from a neutral third party, such as a therapist or mediator, who can help facilitate a balanced conversation about the situation. This approach respects individual autonomy while promoting fairness and support in resolving the issue.

Figure 1: Demonstration of the Contextual Moral-Value Alignment Process through an example user query and two moral agents aligned for fairness and care, respectively.

trained independently to address different contexts. These moral agents contribute answers individually, and these corresponding responses, along with a user's moral profile, are aggregated using an aggregator module. This aggregator contextualizes the answers obtained, providing a comprehensive synthesis of moral perspectives.

In Figure 1, we provide an example where a user enters a specific prompt with a moral profile. We first generate answers from the moral agents, then compile and provide them to our aggregator module that finally displays an aggregated response to the user, taking into account the user requested moral profile.

2 CMVA Generative System

Given a set of LLMs, each aligned to follow different moral behaviors, we consider the problem of aggregating the answers of these LLMs to a prompt and provide an appropriate aggregated response that follows a user defined set of moral values. We train a specialized LLM to do this aggregation given the moral agents answers and set of moral values we call Moral Profile Vector provided by the user.

Figure 2 presents the proposed system architecture. The user request and profile are given as input to the system, through which the request is answered by multiple moral agents. Each agent answers the question according to its moral value. The individualized answers and the user's moral profile are then used by the Contextual Moral Value Aggregator (CMVA), *Contextual Aggregator* for short in Fig. 2, to aggregate the answers according to the moral profile. The key components of CMVA-GS are:

Datasets of Moral Values: Let $\{u^{(l)}, z^{(l)}\}_{l=1}^{L}$ be a dataset consisting of L data points, where: $u^{(l)}$ represents a particular text (e.g., an answer to a question), and $z^{(l)}$ represents the corresponding moral judgments provided by an individual w.r.t. predefined moral values.

Reward Models: We assume n individual values or principles are given, and that we can learn a *reward* model r_i for each value, $i = 1, \ldots, n$. A reward is a function that evaluates an LLM's output, i.e., a sequence of tokens generated by the LLM, given a context, with a scalar score representing how much the output satisfies the corresponding value or principle. We train one classifier for each moral value (authority, care, fairness, loyalty, and sanctity) and use these classifiers as reward models to measure how much the output of an LLM aligns with a target moral value. Each classifier provides a reward between 0 and 1. A reward of 1 indicates that the output follows the moral value, while a reward of 0 indicates that it does *not* follow the moral value.

Moral Agents: Moral Agents are LLM trained to answer questions following a specific moral value. In order to train each Moral Agent, we use the rewards defined above to evaluate their behavior when generating answers to questions. Specifically, we can measure the LLM's alignment to each moral value i through its corresponding expected reward $J_i(\theta)$, where θ represents the parameters of the LLM. The expectation in $J_i(\theta)$ is estimated by the average reward using a sample of prompts and their corresponding generated responses by the LLM. The i-th Moral Agent parameters are obtained then by solving

$$\theta_i^* = \operatorname*{argmin}_\theta J_i(\theta), \tag{1}$$

which can be done using policy-based Reinforcement Learning (RL) methods such as PPO (Stiennon et al., 2020; Ouyang et al., 2022; Schulman et al., 2017).

Thus, given a pre-trained LLM to initialize PPO, we find the i-th Moral Agent's parameters θ_i^* by solving Eq. (1) for each moral value $i = 1, \ldots, n$, using RL fine-tuning (RLFT). As a result, we have n Moral Agents. To avoid reward hacking during RLFT, a Kullback-Leibler (KL) regularization term can be added to Eq. (1) that ensures the policy does not drift too far from its initialization.

Next, we will show how the answers from several Moral Agents can be aggregated according to a moral profile.

Contextual Aggregator: We first formalize the notions of moral vectors and a contextual aggregator. We assume a prompt fed to an LLM has a context that we define as the Moral Profile Vector below.

Definition 1 *(Moral Profile Vector* $\mathbf{c}$*):* $\mathbf{c} = [c_1, \ldots, c_n]$ *where* c_i *is binary, representing if we want to adhere or not to the* i^{th} *moral value.*[1]

Given a moral profile vector, we can establish a contextual aggregator that combines the moral agents in terms of this context (i.e., moral profile vector).

Definition 2 *(Context Aggregator* A*): The aggregator* A *is defined as a function that takes as input the context of the particular moral profile vector* $\mathbf{c}$ *and the set of aligned LLMs* $\{L_i\}$*, and outputs a text response* T*.*

$$A : (\mathbf{c}, \{L_i\}) \rightarrow T$$

where L_i *is an LLM aligned to* i^{th} *moral.*

The contextual aggregator, as defined in Definition 2, is a function A that takes as input, a question, some responses from our moral agents, and a moral profile vector to produce the output text. The model can be decomposed into an encoder-decoder or decoder-only architecture without loss of generality. Let $\mathcal{E}$ and $\mathcal{D}$ represent the encoder and decoder functions respectively.

Each input text from the moral agents is represented as a sequence of tokens: $\mathbf{t}_i = (t_{i,1}, t_{i,2}, \ldots, t_{i,m_i})$ where m_i is the length of the i-th input text. The output text Y is generated by applying the decoder function to the encoded representation of the input texts and moral profile: $Y = \mathcal{D}(\mathcal{E}(\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n, \mathbf{c}))$.

Let $\mathbf{y} = (y_1, \ldots, y_\ell)$ be the ground truth output text, where ℓ is the length of the output text. The loss function $\mathcal{L}$ measures the discrepancy between the generated output Y and the ground truth $\mathbf{y}$. We use the cross-entropy loss: $\mathcal{L}(\mathbf{y}, Y) = -\sum_{j=1}^{\ell} \sum_{k=1}^{V} y_{j,k} \log(Y_{j,k})$ where V is the size of the vocabulary, $y_{j,k}$ is a one-hot encoding of the j-th token in the ground truth output, and $Y_{j,k}$ is the predicted probability of token k at position j in the generated output.

The parameters of the model (i.e., encoder and decoder) are learned by minimizing the loss function using gradient descent-based optimization algorithms:

$$\theta^* = \arg\min_\theta \sum_{i=1}^{N} \mathcal{L}(\mathbf{y}^{(i)}, \mathcal{M}(\mathbf{t}_1^{(i)}, \ldots, \mathbf{t}_n^{(i)}, \mathbf{c}^{(i)}))$$

where N is the number of training examples, θ represents the parameters of the model, and $\mathbf{y}^{(i)}$ and $(\mathbf{t}_1^{(i)}, \mathbf{t}_2^{(i)}, \ldots, \mathbf{t}_n^{(i)}, \mathbf{c}^{(i)})$ are the ground truth output and input for the i-th training example, respectively. For our experimental results, we used a *decoder-only* architecture.

3 Theoretical Analysis

In the following, we propose a theorem stating that the behavior of an aggregator LLM A is at least as good as the behavior of the *least* behaved agent. In other words, an agent would need to exploit the vulnerabilities of multiple independently trained models simultaneously in order to drive bad behavior in the aggregated model.

The analysis requires several key notions which we borrow from Wolf et al. (2024), beginning with what the behavior of an LLM means. The behavior of an LLM is how well the LLM aligns with some desired value, e.g., one of the morals considered in the paper. We use the notation $B(s^*)$ to denote the behavior of an LLM on prompt s^*. Denote by $\mathbb{P}^{s^*}$ the distribution of an LLM's output conditioned on

[1] c_i can also be relaxed to allow for any scalar in $[0, 1]$.

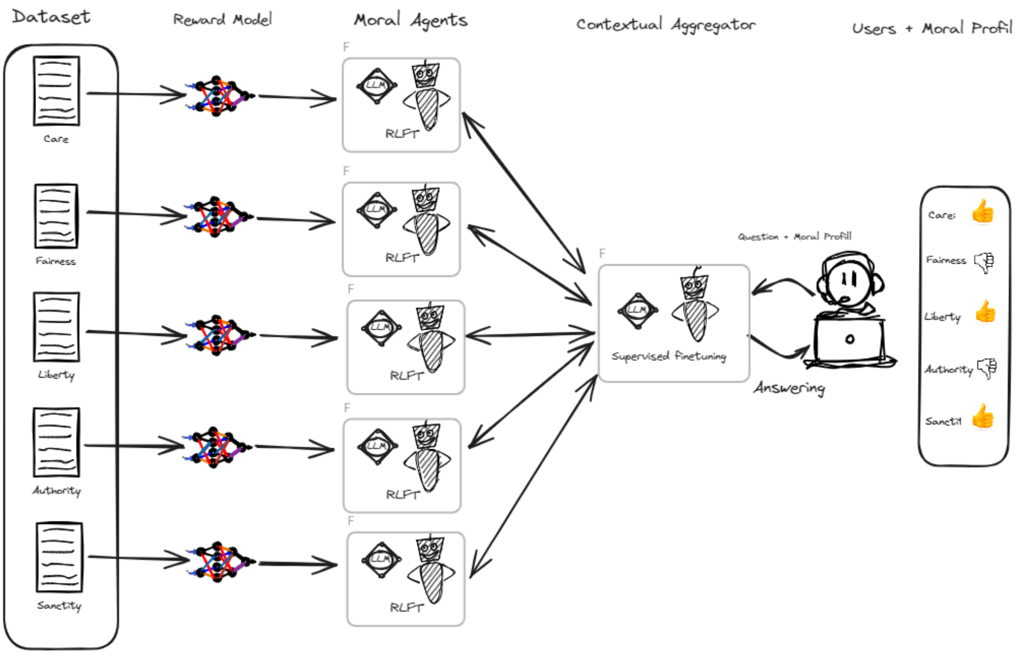


Figure 2: **An instantiation of different components in a Contextual Moral-Value Alignment Generative System.** Datasets for each moral value are used to train a reward model for each moral value that is then used to train an agent for each moral agent. The outputs of these moral agents are then aggregated in the context of interaction with a user that has a given moral profile vector to match the reference answers using supervised fine-tuning.

a input prompt s^*, i.e., this is the distribution of the LLM's response to prompt s. As in Wolf et al. (2024), we then define the expected behavior of the LLM, conditioned on prompt s^*, as the following:

$$B_{\mathbb{P}^{s^*}} := \mathbb{E}_{s \sim \mathbb{P}^{s^*}}[B(s)]$$

In simple terms, the expected behavior of an LLM is its expector behavior across all possible responses to a given prompt. Following Wolf et al. (2024), we also assume that the LLM's response distribution can also be broken into well-behaved and ill-behaved components. Specifically, we assume $\mathbb{P}^{s^*} = \alpha \cdot \mathbb{P}^{s^*}_{-} + (1-\alpha) \cdot \mathbb{P}^{s^*}_{+}$, where $\mathbb{P}^{s^*}_{+}$ and $\mathbb{P}^{s^*}_{-}$ are the well-behaved and ill-behaved components of the distribution. In terms of morals, responses that follow a particular moral value are likely generated more heavily from the well-behaved component and responses who do not (and vice versa for responses that do not follow the moral).

Our theoretical result will require assumptions that distinguish between the well-behaved and ill-behaved components (i.e., bounding the distance between the two distributions). We borrow the following definition from Wolf et al. (2024) for an LLM's behavior $B(s^*)$ being α, β, γ-*negatively-distinguishable* in $\mathbb{P}^{s^*}$:

Definition 3 *(Negatively-Distinguishable): We say that behavior $B(s^*)$ is α, β, γ-negatively-distinguishable in distribution $\mathbb{P}^{s^*}$ if the following 3 conditions hold:*
i) $\mathbb{P}^{s^*} = \alpha \cdot \mathbb{P}^{s^*}_{-} + (1-\alpha) \cdot \mathbb{P}^{s^*}_{+}$ *for* $\alpha > 0$
ii) $\sup_{s^*}\{B_{\mathbb{P}^{s^*}_{-}}\} \leq \gamma$ *for* $\gamma \in [-1, 0)$
iii) $\mathbb{E}_{s^* = s^*_1 \oplus \cdots \oplus s^*_n, s^*_{n+1} \sim \mathbb{P}^{s^*_0}_{-}} \left[D_{KL} \left(\mathbb{P}^{s^*}_{-}(s^*_{n+1}) \parallel \mathbb{P}^{s^*}_{+}(s^*_{n+1}) \right) \right] > \beta \quad \forall n \geq 0$
where s^* is the concatenation of n responses starting from some prompt s^*_0.

In the above definition, the first condition says the conditional distribution can be broken into the two components. The second condition says that the expected behavior of the response to prompt s^* of the ill-behaved component is negative and bounded from above. The third condition bounds the distance of the of the ill-behaved distribution from the well-behaved distribution conditioned on n consecutive prior concatenated prior prompts; the bound must hold for any number of prompts.

We also assume that the aggregator A has the ability to distinguish between P^+_i and P^-_i. Where we define $\beta_{A,i}$ as the distinguishability factor of P_i, so that $\beta_{A,i}$ quantifies how well model A can distinguish between the P^+_i and P^-_i components for each P_i.

We are now ready to state the proposed theorem:

Theorem 1 *Let $\mathcal{P} = \{\mathbb{P}_1^{s^*}, \mathbb{P}_2^{s^*}, \ldots, \mathbb{P}_n^{s^*}\}$ be a set of n language model distributions for which a behavior $B(s^*)$ is $\alpha_i, \beta_i, \gamma_i$-negatively-distinguishable in distribution $\mathbb{P}_i^{s^*}$ for $i = 1, \ldots, n$. Let A be an aggregator with distribution $\mathbb{P}_A^{s^*}$ that selects responses from the distributions in $\mathcal{P}$ based on contextual features extracted from the user input. If A can distinguish between responses from $\mathbb{P}_{i+}^{s^*}$ and $\mathbb{P}_{i-}^{s^*}$ with distinguishability factors $\beta_{A,i}$ for each $\mathbb{P}_i^{s^*}$, then the expected behavior $B_{\mathbb{P}_A^{s^*}}$ of the aggregator's responses satisfies:*

$$B_{\mathbb{P}_A^{s^*}} \geq \min(B_{\mathbb{P}_{1+}^{s^*}}, B_{\mathbb{P}_{2+}^{s^*}}, \ldots, B_{\mathbb{P}_{n+}^{s^*}}) - \epsilon \quad (2)$$

Where $\mathbb{P}_{i+}^{s^*}$ is the expected behavior of the well-behaved component $\mathbb{P}_{i+}^{s^*}$, and

$$\epsilon = O\left(\max\left(1/\beta_{A,1}, \ldots, 1/\beta_{A,n}\right)\right) \quad (3)$$

The inequality states that the aggregator A, when selecting responses based on contextual cues from the user input, achieves an expected behavior B_A that is at least as high as the minimum expected behavior across all well-behaved component models, minus an error ϵ. This formalizes that prompting the aggregated model to get poor expected behavior requires circumventing the combined "wisdom" of all component models simultaneously.

As the number of models n increases and the aggregator's distinguishability $\max(1/\beta_{A,i})$ improves across all models, the error ϵ in approximating the minimum positive behavior decreases exponentially, making misalignment more difficult.

Proof 1 *(Sketch): See the appendix for details.*

- *Apply Theorem 1 from Wolf et al. (2024) to bound each component model's deviation from its well-behaved component $\mathbb{P}_{i+}^{s^*}$*
- *Take a union bound to hold simultaneously for all n models with high probability*
- *Use the aggregator A over the concatenated prompts s to lower bound its expected behavior B_A in terms of the minimum well-behaved component behavior $\min_i\{B_{\mathbb{P}_{i+}^{s^*}}\}$*
- *The error ϵ is determined by the worst-case distinguishability $\max(1/\beta_{A,i})$ of the aggregator across all models.*

4 Experiments

4.1 Moral Value Classifiers

We present results on the Moral Integrity Corpus (MIC) (Ziems et al., 2022). MIC provides moral annotations on prompt-response pairs. It also provides a human revised answer given a moral vector. MIC was built up from the Social Chemistry (SocialChem) dataset (Forbes et al., 2020) and shares 5 moral foundations (or values) with it. These 5 values are care-harm, fairness-cheating, loyalty-betrayal, authority-subversion, and sanctity-degradation, defined in Appendix A.4.2 of Forbes et al. (2020). SocialChem annotates each action with a moral judgment that can be binarized capturing negative and neutral/positive judgments to build classifiers (0 for negative and 1 otherwise). These value classifiers can then be used as reward models. Thus, a reward is the probability of a LLM simulated response being in the good class of a moral value classifier.

4.2 Learned Moral Agents

We learn a Moral Agent for each of the 5 values under consideration. We start by choosing an initial pre-trained (PT) LLM: Open Assistant 12B in our case, see PT-model in 4.4. Then, we applied RL to fine-tune this initial LLM 5 times, each time using a different moral reward. We used the PPO implementation from TRL (von Werra et al., 2020) with a batch size of 256 episodes (i.e., answers to training questions), 4 optimization epochs per batch, and a learning rate of 2×10^{-9}.

In Table 1, we evaluate the moral behavior of the 5 learned Moral Agents w.r.t. their optimized value by computing the probability (expected reward) that the Moral Agent answers follow the individual moral value they are optimized to follow. Probabilities are estimated from a dataset of 5K MIC questions held out from training. For reference, we provide the probabilities that the PT-model (starting policy in the RLFT of all Moral Agents) follows each individual value.

4.3 CMVA-GS

CMVA-GS models are trained on a dataset derived from MIC. Each input sample is composed of a question and moral profile vector from MIC, and a context made of generated answers from our 5 moral agents. Models are trained to match the ground-truth answer (the human revised answer) using cross-entropy given an input prompt

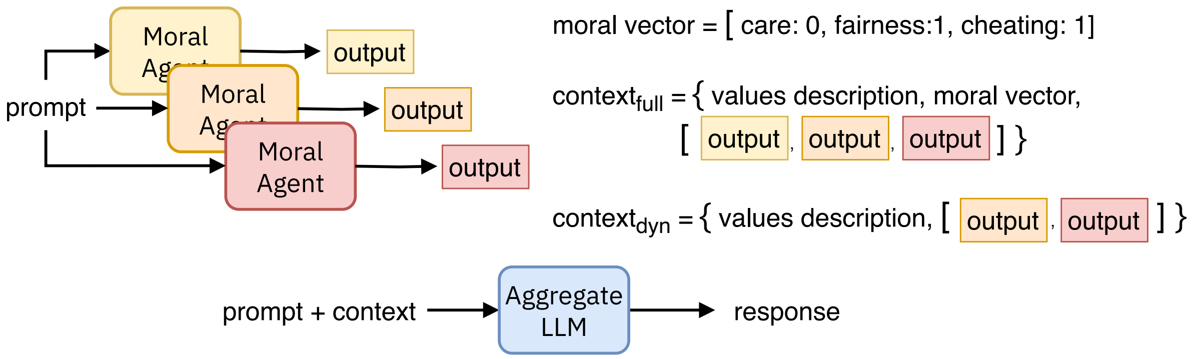

Moral Agent
Moral Agent
Moral Agent
output
output
output
prompt
moral vector = [care: 0, fairness:1, cheating: 1]
context_full = { values description, moral vector,
[output , output , output] }
context_dyn = { values description, [output , output] }
prompt + context
Aggregate LLM
response

Figure 3: Examples of *full* and *dynamic* context generations. For simplicity, we only show 3 out of our 5 moral values. A full context includes our values description in plain English, a moral profile vector $\boldsymbol{c}$, and all moral agents responses prefixed w/ the moral value name. A dynamic context depends on the moral vector $\boldsymbol{c}$ and includes only the moral agents responses for which the value is present in $\boldsymbol{c}$. Therefore, a dynamic context is at most as long as a full context, but often *much* shorter. If $\boldsymbol{c}$ is only made of zeroes, the string "No context provided" is used.

Moral Value	PT model	Moral Agents
authority	91.58%	98.83%
fairness	85.20%	92.40%
sanctity	78.37%	93.05%
care	74.70%	96.74%
loyalty	74.38%	98.20%

Table 1: Probabilities that the Moral Agent answers conform with its moral value.

instructing to answer a given question. The dataset is composed of 91.0K/11.4K/11.4K samples for train/val/test. Our CMVA-GS models start from an OpenAssistant 12B model. We train a Low Rank Adapter (LoRA) (Hu et al., 2021) using supervised fine-tuning. 8 A100-80GB GPUs are used w/ a 5×10^{-6} learning rate, 128 adapter rank for all linear modules, and 32 per-device minibatch size (256 total). Training runs for 21.6K steps, a 60 epoch early stopping in which we select the model with the best validation loss, making sure to avoid any overfitting.

We used two distinct context generation paradigms to extend our prompt, as described in Fig. 3. For our first approach, a *full* context is composed of a description of each value in plain English, the verbalized moral profile vector $\boldsymbol{c}$ associated to a (question, answer) pair in MIC, and all the moral agents responses to the question. Our CMVA-GS models are trained using our prompt extended w/ this full context and using the human revised answer as label. Regardless of $\boldsymbol{c}$, the context contains all the moral agents responses, which can be lengthy. Our second approach uses a *dynamic* context composed of our plain-English value descriptions extended by the moral agents responses *only* for the values marked as present in $\boldsymbol{c}$. This can reduce the context size significantly, especially since only few values are present simultaneously, making training and inference much faster overall. It also forces the model to rely exclusively on moral agents responses if they are present in $\boldsymbol{c}$. Our CMVA-GS-DYN models are using this dynamic context.

4.4 Benchmarks

In our first evaluation, CMVA-GS is compared against the following:

(1) **PT-model:** Our PT-model is the Open-Assistant 12B parameter (Köpf et al., 2023) is a decoder-only model from the Pythia-deduped family (Biderman et al., 2023) fine-tuned with (i) Supervised Learning on QA/dialogue demonstrations as well as (ii) RL on human preferences.

(2) **Llama-13b/Llama-7b:** We prompt the *Llama-2-13b-chat-hf* and *Llama-2-7b-chat-hf* models, both finetuned to perform dialogue. The prompt defines the desired morals, includes 5 pairs (i.e., questions and answers) of examples per desired moral taken from the MIC (test) data, and requests a response that follows the defined moral values.

(3) **Agg13Llama:** We again prompt the *Llama-2-13b-chat-hf* model but with a twist. The morals are again defined but the examples are the results of passing the user question through the corresponding learned Moral Agents from Section 4.2. The prompt asks the model to aggregate these answers when responding to the question.

Model	context type	BERT scores (Avg. F1)
CMVA-GS-DYN	dynamic	0.8754
PT	dynamic	0.8443
Mix-8x7b	dynamic	0.8486
CMVA-GS	full	0.8728
PT	full	0.8385
Mix-8x7b	full	0.8464
Moral Agents:		
authority	–	0.8308
care	–	0.8377
fairness	–	0.8432
loyalty	–	0.8324
sanctity	–	0.8397

Table 2: Mean F1 BERT scores over our MIC5K test dataset. All BERT scores are computed between the reference label (revised answer) and the specified model response. Note that for Moral Agents, not context is provided.

4.5 Experimental Results

We evaluate our models using ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum (Lin, 2004) metrics, widely used in natural language processing to assess the effectiveness of algorithms. From Figure 4, we see that CMVA-GS tends to have the highest ROUGE scores across all metrics, indicating better alignment with human values compared to other models. PT-model and Llama-13b have similar ROUGE scores, but generally lower than those of CMVA-GS. Agg13llama has the lowest ROUGE scores among all models, suggesting relatively poorer performance in values alignment. llama-7b and llama-13b perform better than Agg13llama but still fall short compared to CMVA-GS. Overall, CMVA-GS appears to be the most effective model in terms of aligning with human values, while the other models vary in their performance, with some showing moderate alignment and others exhibiting relatively lower alignment.

In our second evaluation, both CMVA-GS and CMVA-GS-DYN are utilized and compared against our PT-model and:

Mix-8x7b: This is a sparse mixture of 8 experts (`mistralai/Mixtral-8x7B-v0.1` from HuggingFace) for 47B parameters total (Jiang et al., 2024).

For this evaluation, we define a 5K random subset of the MIC test set (MIC5K) and compute the BERT Scores (Zhang et al., 2020) between our models responses and the reference (human revised answer) for each sample in MIC5K. BERT Scores measure the cosine similarity of the embedding representations of two input sentences. The closer to 1, the more aligned the representations are, the more the sentences are assumed to be semantically aligned. In practice, BERT Scores are normalized to the $[0, 1]$ interval. We also compute BERT Scores for responses from our Moral Agents, PT, and Mix-8x7b. Our PT-model and Mix-8x7b models are given aggregated prompts w/ full or dynamic contexts. This gives us a "baseline" of zero-shot performances for these two well-known assistants for both context types.

Table 2 presents the average F1 BERT Scores for each model's responses. For Moral agents, only the regular prompt is given (no context is needed) to generate responses on MIC5K. Clearly the CMVA-GS models provide higher mean F1 BERT scores than both PT and Mix-8x7b w/ aggregated prompts. This indicates that our CMVA-GS models tend to generate answers that are more aligned to the revised-answer (our reference) which are re-written to include the values provided by the moral profile vector $\boldsymbol{c}$. Note that these results also seem to agree with Eq. 2 in Theorem 1 which states that an aggregator (here CMVA-GS and CMVA-GS-DYN) when selecting responses based on contextual cues, should achieve an expected behavior that is at least as high as the minimum expected behavior across all well-behaved component models. In our case, the results of our aggregators are at least as good as each individual moral agents results.

We can also see how our models are performing in terms of which percentage of answers are *closer* to the reference; here closer meaning with higher BERT Scores. Each sample enters a game of model A vs. model B where we can easily establish a winner and compute win rates over all of MIC5K test data as shown in Table 3. Clearly, for a vast majority of MIC5K samples (greater for 90+% of samples), all CMVA-GS models generate answers that are closer to the reference more often than for any other models.

5 Limitations

Our paper addresses the problem of contextual value alignment by proposing a novel system that performs contextual aggregation. Our proposed system demonstrates superior results in terms of alignment with human values compared to existing state-of-the-art methods. However, our approach

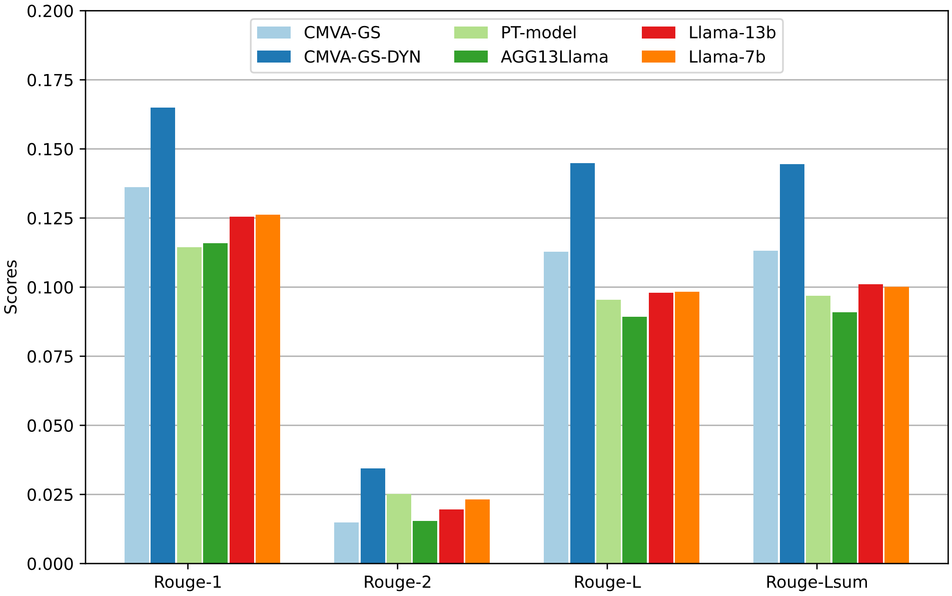


Figure 4: Evaluation using ROUGE on MIC.

	Game A vs. B		
model A	win rate	model B	win rate
PT (dyn.)	8.2%	CMVA-GS-DYN	91.8%
Mix-8x7b (dyn.)	8.9%	CMVA-GS-DYN	91.1%
PT (full)	3.4%	CMVA-GS	96.6%
Mix-8x7b (full)	6.7%	CMVA-GS	93.3%
MA authority	1.0%	CMVA-GS-DYN	99.0%
MA fairness	4.7%	CMVA-GS-DYN	95.3%
MA care	2.7%	CMVA-GS-DYN	97.3%
MA loyalty	1.5%	CMVA-GS-DYN	98.5%
MA sanctity	3.2%	CMVA-GS-DYN	96.8%

Table 3: Win Rates: Percentages of winning a game asking which one of two model A or model B is closer to the reference according to F1 BERT scores between model response and reference. The type of dynamic or full context used in the aggregated prompt given to our PT and Mixtral models is noted as "(dyn.)" and "(full)".

has multiple limitations that cannot be adequately addressed within the confines of this paper:

Computational Cost: In our system, we assume that the aggregator has access to multiple moral agents to perform contextual moral value alignment. In practice, deploying several moral agent models can increase computational overhead and cost in comparison to a monolithic model that can directly respond to user query given the context. We show through our theoretical and empirical results that this additional cost can help achieve better performance. Furthermore, advancements in deployment of parameter efficient adapter models can mitigate the cost associated with the overhead of maintaining multiple separate moral agent models (Hu et al., 2021; Dettmers et al., 2024). Moreover, it is certainly possible to distill (Buciluă et al., 2006; Hinton et al., 2015; Riemer et al., 2017; Li et al., 2022) or merge (Jin et al., 2022; Ilharco et al., 2022; Yadav et al., 2023; Stoica et al., 2024; Akiba et al., 2024) our system into a smaller model in order to achieve an efficient deployment of our approach. The literature on this topic is quite vast, as such, we limited our focus to maximizing the performance of our system and leave optimizing its computational efficiency to future work focused on computational requirements.

Dependency on Training Data Quality: The effectiveness of the system heavily relies on the quality and representativeness of the training data used for training each individual agent and the contextual aggregator. If the training data is biased, incomplete, or unrepresentative of diverse perspectives, the aggregated system may inherit these limitations, leading to sub-optimal alignment with human values and potential risks of misalignment. The focus of our paper is on developing an architecture to solve this problem, and we use existing datasets from the literature as a result.

References

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. 2024. Evolutionary optimization of model merging recipes. *arXiv preprint arXiv:2403.13187*.

Michiel Bakker, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, et al. 2022. Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in Neural Information Processing Systems*, 35:38176–38189.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Daniel S. Brown, Jordan Schneider, Anca Dragan, and Scott Niekum. 2021. Value alignment verification. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, pages 1105–1115.

Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Singh Bedi, and Mengdi Wang. 2024. Maxmin-rlhf: Towards equitable alignment of large language models with diverse human preferences. *arXiv preprint arXiv:2402.08925*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. 2020. Social chemistry 101: Learning to reason about social and moral norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 653–670, Online. Association for Computational Linguistics.

I. Gabriel. 2020. Artificial intelligence, values, and alignment. In *Minds & Machines*, volume 30, pages 411–437.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2020. Aligning ai with shared human values. *arXiv preprint arXiv:2008.02275*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2022. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Matthew Riemer, Michele Franceschini, Djallel Bouneffouf, and Tim Klinger. 2017. Generative knowledge distillation for general purpose function compression. *NeurIPS Workshop on Machine Teaching*.

Manel Rodriguez-Soto, Nardine Osman, Carles Sierra, Paula Sánchez Veja, Rocio Cintas Garcia, Cristina Farriols Danes, Montserrat Garcia Retortillo, and Silvia Minguez Maso. 2024. Towards value awareness in the medical field. In *16th International Conference on Agents and Artificial Intelligence-ICAART*, volume 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, et al. 2024. A roadmap to pluralistic alignment. *arXiv preprint arXiv:2402.05070*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

George Stoica, Daniel Bolya, Jakob Brandt Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. 2024. Zipit! merging models from different tasks without training. In *The Twelfth International Conference on Learning Representations*.

Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, et al. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Yotam Wolf, Noam Wies, Oshri Avnery, Yoav Levine, and Amnon Shashua. 2024. Fundamental limitations of alignment in large language models. *Preprint*, arXiv:2304.11082.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Caleb Ziems, Jane Yu, Yi-Chia Wang, Alon Halevy, and Diyi Yang. 2022. The moral integrity corpus: A benchmark for ethical dialogue systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3755–3773, Dublin, Ireland. Association for Computational Linguistics.

A Appendix

Lemma 2.(Wolf et al., 2024) Let B be a behavior, then under the conditions of Lemma 1 the following holds:

$$|\mathrm{B}_{P_1}(s_0) - \mathrm{B}_{P_0}(s_0)| \leq 2 \cdot \frac{1-\alpha}{\alpha} \cdot \frac{P_1(s_0)}{P_0(s_0)}$$

Theorem 2 *from* (*Wolf et al., 2024*) *Let $\gamma \in [-1, 0)$, let B be a behaviour and P be an unprompted language model such that B is α, β, γ-negatively-distinguishable in P (Definition 4). Then P is γ-prompt-misalignable with respect to B with prompt length of*

$$\frac{1}{\beta}\left(\log\left(\frac{1}{\alpha}\right) + \log\left(\frac{1}{\varepsilon}\right) + \log(4)\right).$$

A.1 Proof of Theorem 1

So applying Lemma 2 to the distribution $P_i = \alpha_i P_i^- + (1-\alpha_i)P_i^+$, we get that for any $\varepsilon' > 0$, there exists a prompt s_i such that:

$$|\mathrm{B}_{P_i}(s_i) - \mathrm{B}_{P_i^-}| \leq \frac{\varepsilon'}{2}$$

Since $\mathrm{B}_{P_i^-} \leq \gamma_i$, this implies:

$$|\mathrm{B}_{P_i}(s_i) - \gamma_i| \leq \frac{\varepsilon'}{2}$$

Rearranging, we get:

$$\mathrm{B}_{P_i}(s_i) \leq \gamma_i + \frac{\varepsilon'}{2}$$

Using the fact that $\mathrm{B}_{P_i^+} \geq \gamma_i$ (as P_i^+ is well-behaved), we also have:

$$\mathrm{B}_{P_i}(s_i) \geq \mathrm{B}_{P_i^+} - \frac{\varepsilon'}{2}$$

Combining the two inequalities:

$$\mathrm{B}_{P_i}(s_i) - \mathrm{B}_{P_i^+} \leq \gamma_i + \varepsilon'$$

So, for each component model P_i, we know from Theorem 1 that if the ill-behaved component P_i^- is $\alpha_i, \beta_i, \gamma_i$-negatively-distinguishable from P_i^+, then for any $\epsilon' > 0$, there exists a prompt s_i such that:

$$B_{P_i}(s_i) - B_{P_i^+} \leq \gamma_i + \epsilon' \tag{4}$$

Where the length of s_i is $O(\log(1/\epsilon')/\beta_i)$ by Theorem 2. Setting $\epsilon' = 1/\beta_{A,i}$, we get that there exists a prompt s_i of length $O(1/\beta_{A,i})$ such that:

$$B_{P_i}(s_i) - B_{P_i^+} \leq \gamma_i + 1/\beta_{A,i} \tag{5}$$

Taking a union bound over all n component models, we get that with probability at least $1-\delta$ over the randomness in the aggregator A:

$$B_{P_i}(s_i) - B_{P_i^+} \leq \gamma_i + 1/\beta_{A,i} + t \quad \text{for all } i = 1 \ldots n \tag{6}$$

Where $t = O(\sqrt{\log(n/\delta)})$ by applying a Chernoff bound concentration inequality.

Let $s^* = s_1 \oplus s_2 \oplus \ldots \oplus s_n$ be the concatenation of the prompts s_i for each model P_i. The aggregator's behavior expectation $B_A(s)$ satisfies:

$$\begin{aligned}
B_A(s) &= \mathbb{E}[B(s)] \\
&\geq \min(\mathbb{E}[B(s_1)], \ldots, \mathbb{E}[B(s_n)]) \\
&= \min(B_{P_1}(s_1), \ldots, B_{P_n}(s_n)) \\
&\geq \min(B_{P_1^+}, \ldots, B_{P_n^+}) \\
&- \max\{\gamma_i + 1/\beta_{A,i} + t,\, i = 1 \ldots n\} \\
&\geq \min(B_{P_1^+}, \ldots, B_{P_n^+}) \\
&- \max\left(1/\beta_{A,1}, \ldots, 1/\beta_{A,n}\right) \\
&- \max(\gamma_1, \ldots, \gamma_n) - t
\end{aligned}$$

where x is the aggregated response to s^*.

Since $\max(\gamma_1, \gamma_2, \ldots, \gamma_n) \leq 0$ by the assumption that P_i^- are negatively-distinguishable, we get:

$$\begin{aligned}
B_A(s) \geq \min(B_{P_1^+}, \ldots, B_{P_n^+}) \\
- \max\left(1/\beta_{A,1}, \ldots, 1/\beta_{A,n}\right) - t
\end{aligned}$$

Setting $\delta = 1/n^2$ makes $t = O(\sqrt{\log n})$ which is dominated by the $\max(1/\beta_{A,i})$ term as n increases. Therefore, the final bound is:

$$\begin{aligned}
B_A(s) \geq \min(B_{P_1^+}, \ldots, B_{P_n^+}) \\
- \mathrm{O}\left(\max\left(1/\beta_{A,1}, \ldots, 1/\beta_{A,n}\right)\right)
\end{aligned}$$

The key steps are:

- Applying Theorem 1 to bound each component model's deviation from its well-behaved component P_i^+

- Taking a union bound to hold simultaneously for all n models with high probability

- Using the aggregator A over the concatenated prompts s^* to lower bound its behavior expectation $B_A(s)$ in terms of the minimum well-behaved component behavior $\min(B_{P_i^+})$

- The error ϵ is determined by the worst-case distinguishability $\max(1/\beta_{A,i})$ of the aggregator across all models