

Learning from Critiques: A Geometric Framework for Response Improvement

Stanford CS224N Custom Project

Haozhan Gao

Stanford Graduated School of Business

haozhan.gao@stanford.edu

Abstract

Current language model post-training pipelines rely heavily on pairwise preference rankings or fully written target responses, both of which are expensive to collect from human annotators. In this project, we study a cheaper and more natural supervision source: natural-language critiques. We propose a geometric framework that learns a response encoder $e(x, y)$ and a critique encoder $g(x, y, c)$, where the critique vector is trained to represent a direction from a response toward the ideal reference response. This formulation allows critiques to serve as structured supervision without requiring explicit pairwise rankings or gold rewritten answers. Empirically, our critique-based method achieves a test ranking accuracy of 0.738 when using critique norm as a quality proxy, outperforming a simple RLHF reward-model baseline that achieves 0.67. These results suggest that critiques contain rich and scalable alignment signal, and that critique-based supervision can be a promising alternative or complement to standard preference-based reward learning.

1 Key Information to include

- Mentor: Fang Wu
- External Collaborators (if you have any): None
- Sharing project: No

2 Introduction

Large language models are typically post-trained with two main forms of human supervision. In supervised fine-tuning (SFT), annotators are asked to write high-quality target responses for a prompt. In reinforcement learning from human feedback (RLHF), annotators are instead asked to compare two responses and indicate which one is better. Both forms of supervision are expensive. Writing an ideal answer requires substantial expertise and time, especially for open-ended or technical prompts. Pairwise preference annotation is often cheaper, but it is still labor-intensive, and in practice annotators may not read both responses carefully or may provide only weak and noisy signal. By contrast, natural-language critiques are abundant in everyday interactions and are often much easier for humans to provide: instead of writing the best answer from scratch or comparing two long responses side by side, a user can simply say what is missing, unclear, or incorrect.

This project studies whether such critiques can be turned into a useful learning signal for language model alignment. The core idea is geometric. Let $e(x, y)$ denote an embedding of response y under prompt x , and let $g(x, y, c)$ denote an embedding of a critique c applied to response y . We would like $g(x, y, c)$ to behave like an improvement vector: if a critique correctly describes how a response should be improved, then a better response y' should lie closer to the critique-implied target point $e(x, y) + g(x, y, c)$. This gives critiques both directional meaning, by indicating how a response should change, and magnitude meaning, by indicating how far the response may be from an improved or ideal answer. In our setup, a *local critique* describes why the losing response is worse than the winning one and therefore captures the edit from loser to winner, while a *global critique* evaluates a single response against a broader notion of

what an ideal answer should look like. The training objective therefore combines three pieces: a local edit term that aligns a local critique with the observed winner-minus-loser displacement, a geometric consistency term that couples local and global critiques, and a magnitude-gap term that calibrates critique norms using human preference strength.

To evaluate this idea, we use HelpSteer2 preference data augmented with local and global critiques, and train two encoders: a response encoder $e(\cdot)$ and a critique encoder $g(\cdot)$. In the final implementation, we use the Qwen3-Embedding-0.6B backbone with LoRA on the query and value projections, together with separate MLP heads for responses and critiques. The training pipeline constructs five text objects per example—winning and losing responses, a local critique of the loser, and global critiques of both winner and loser—and optimizes the critique-conditioned geometric objective over train, validation, and test splits.

Empirically, the learned critique geometry is informative. On the test set, the norm of the inferred global critique vector is a useful proxy for response quality: responses with shorter critique norms are more likely to be preferred. This suggests that critiques can encode not only qualitative feedback but also a meaningful notion of distance to improvement. More broadly, the contribution of this project is not merely a new scalar reward model, but a framework for converting short natural-language critiques into structured supervision for alignment. If successful, such a framework could reduce reliance on pairwise rankings and fully written target responses, making post-training more scalable and less costly for human experts.

3 Related Work

Our work is related to several neighboring lines of research in language model alignment and learning from feedback, but it differs from each of them in both supervision format and modeling objective.

The dominant paradigm for post-training large language models is supervised fine-tuning (SFT) followed by preference-based alignment. InstructGPT (Ouyang et al. [2022]) established the now-standard pipeline of training on human-written demonstrations, fitting a reward model from pairwise comparisons, and then optimizing the policy against that reward model using RLHF. More broadly, recent surveys characterize RLHF (Kaufmann et al. [2024]) as a framework built around scalar preference modeling from chosen–rejected pairs. These methods have been highly successful, but they rely on costly supervision: SFT requires annotators to write high-quality target responses, while RLHF requires humans to compare multiple responses carefully, often for long and nuanced prompts.

A closely related line of work tries to reduce the dependence on human preference labels by replacing them with AI feedback. Constitutional AI (Bai et al. [2022]) uses natural-language principles to generate critiques and revisions, and then uses this feedback to train safer assistants, effectively showing that critique-like natural-language supervision can substitute for some human preference judgments. RLAIF (Lee et al. [2023]) further explores replacing human comparisons with AI-generated feedback at scale. These works share our belief that richer and cheaper natural-language feedback can be useful for alignment, but their primary goal remains to improve a standard scalar reward or policy-training pipeline. In contrast, our method does not begin by assuming a scalar reward model $r(x, y)$; instead, it learns a critique-conditioned geometric representation in which critiques act as improvement directions in response space.

Another relevant thread studies critique-and-revise procedures at inference time. Self-Refine (Madaan et al. [2023]) shows that an LLM can generate feedback on its own output and iteratively improve it, without additional task-specific training. This is conceptually close to our motivation: a critique contains information about how an answer should change. However, Self-Refine is primarily an iterative generation-time procedure, whereas our goal is to learn a reusable representation from critique data itself. Rather than using critiques only to revise one output at a time, we try to encode them as structured signals that can supervise representation learning across many examples.

The nearest technical neighbors are recent methods that augment reward models with critiques. Critique-out-Loud (CLOUD) reward models (Ankner et al. [2024]) first generate a natural-language critique and then use it to improve scalar reward prediction. Improving Reward Models with Synthetic Critiques (Ye et al. [2025]) and Critic-RM (Yu et al. [2025]) similarly show that synthetic or self-generated critiques can improve reward-model accuracy and data efficiency. These papers are important because they support the general claim that critiques carry richer signal than binary preferences alone. However, they still use critiques mainly as auxiliary evidence for predicting a scalar reward. Our work differs in the object being learned: instead of treating a critique as textual rationale for a score, we train a critique encoder $g(x, y, c)$ so that a critique behaves like a vector pointing from a response toward a better one. In other words, prior critique-augmented reward models remain fundamentally scalar, whereas our formulation is explicitly geometric.

Our work is also related in spirit to process supervision and process reward models, which argue that richer intermediate feedback can be more informative than final outcome labels alone. For example, process-supervised reward models for mathematical reasoning (Lightman et al. [2023]) show that supervising intermediate reasoning steps can yield better

reward estimates than outcome-only supervision. Although that literature focuses on reasoning trajectories rather than natural-language critiques of responses, it shares our broader motivation: scalar end labels may discard useful structure, and richer feedback can provide a better learning signal. Our contribution can be seen as extending this idea from process annotations to critique annotations.

Finally, our work is motivated by scalable feedback collection. Existing alignment pipelines usually rely on pairwise rankings or full target rewrites. We instead study critiques as a supervision interface and ask a different modeling question: not how to use critiques to better predict a scalar reward, but how to represent them as structured improvement directions in response space. In that sense, our project lies between preference-based alignment and critique-guided refinement.

4 Approach

Our goal is to learn from natural-language critiques without requiring either pairwise preference labels alone or fully written gold responses. We formalize each training example around three objects: a prompt x , a response y , and a critique c . Intuitively, the response embedding should encode where the current answer lies in latent response space, while the critique embedding should encode how that answer should move in order to improve.

4.1 Setup

For a prompt x , suppose we observe two candidate responses y_w and y_ℓ , where y_w is preferred to y_ℓ . We also observe three critiques:

- a **local critique** c_{local} , which explains why the losing response y_ℓ is worse than the winning response y_w ;
- a **global critique** c_w , which evaluates the winning response relative to an ideal answer;
- a **global critique** c_ℓ , which evaluates the losing response relative to an ideal answer.

We learn two encoders:

$$e(x, y) \in \mathbb{R}^d, \quad g(x, y, c) \in \mathbb{R}^d.$$

Here, $e(x, y)$ is the embedding of response y under prompt x , and $g(x, y, c)$ is the embedding of critique c applied to response y . We interpret $g(x, y, c)$ as an *improvement vector*: it should point from the current response toward a better one.

This induces the following geometric picture. If a critique correctly describes how a response should change, then the corrected point

$$e(x, y) + g(x, y, c)$$

should move closer to an improved, and ideally near-optimal, response for prompt x . In this view, the direction of g captures *how* the response should change, while its norm captures *how far* the response is from a better answer.

For notational convenience, define

$$\begin{aligned} e_w &:= e(x, y_w), & e_\ell &:= e(x, y_\ell), \\ g_{\text{local}} &:= g(x, y_\ell, c_{\text{local}}), & g_w &:= g(x, y_w, c_w), & g_\ell &:= g(x, y_\ell, c_\ell). \end{aligned}$$

4.2 Geometric Training Objective

We train the response and critique encoders by combining three losses:

$$\mathcal{L} = \lambda_{\text{local}} \mathcal{L}_{\text{local}} + \lambda_{\text{geom}} \mathcal{L}_{\text{geom}} + \lambda_{\text{mag}} \mathcal{L}_{\text{mag}}.$$

Each term enforces a different aspect of the intended geometry.

4.2.1 Local edit loss

The local critique compares the losing response directly against the winning one, so it should approximate the displacement from loser to winner:

$$\mathcal{L}_{\text{local}} = \|g_{\text{local}} - (e_w - e_\ell)\|_2^2.$$

The intuition is straightforward: if the local critique says what is wrong with y_ℓ relative to y_w , then its embedding should match the edit needed to move from the losing response toward the winning response. This is the most direct supervision signal in the model.

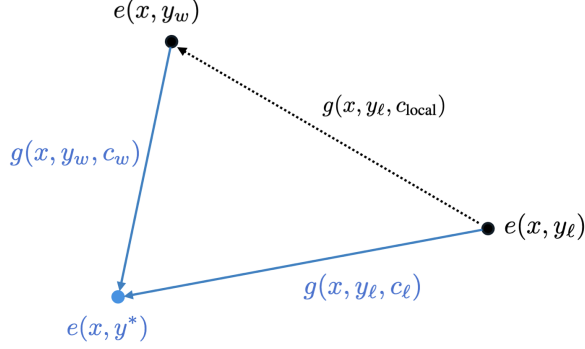


Figure 1: The geometric structure of responses and critiques

4.2.2 Geometric consistency loss

The global critiques of the winner and loser should both point toward roughly the same latent ideal answer. Therefore, after applying the corresponding critique vectors, the two corrected responses should land near one another:

$$\mathcal{L}_{\text{geom}} = \|(e_w + g_w) - (e_l + g_l)\|_2^2.$$

This term does not require us to observe the true ideal response y^* . Instead, it uses the weaker but practical assumption that if both critiques are meaningful and are from the same commentator, then they should move the winner and loser toward a shared high-quality region of response space.

4.2.3 Magnitude calibration loss

In addition to direction, we want critique vectors to carry magnitude information. A worse response should generally require a larger correction than a better one. Let $r \geq 0$ denote the observed preference strength between the winner and loser. We impose the constraint

$$\|g_l\|_2^2 - \|g_w\|_2^2 \approx \alpha r,$$

with α being a tunable hyperparameter. It leads to the loss

$$\mathcal{L}_{\text{mag}} = (\|g_l\|_2^2 - \|g_w\|_2^2 - \alpha r)^2.$$

The role of this term is to weakly calibrate critique norms. If the losing response is much worse than the winning response, then its critique should have larger norm, reflecting a greater distance to improvement. The scalar α controls how preference strength maps into a target gap in squared norms.

4.3 Why this objective is useful

The three losses play complementary roles. $\mathcal{L}_{\text{local}}$ anchors local critiques to observed winner-minus-loser edits, $\mathcal{L}_{\text{geom}}$ uses global critiques to define a shared high-quality region without observing an explicit ideal answer, and $\mathcal{L}_{\text{mag}}$ calibrates critique norms so they can serve as a quality proxy at test time.

5 Experiments

5.1 Data

We use the preference subset of HelpSteer2, a human-feedback dataset containing paired responses and preference annotations for assistant prompts.¹ In particular, we use the fields `prompt`, `response_1`, `response_2`, and `preference_strength`. Each example contains two candidate responses to the same prompt together with a discrete preference-strength label indicating how strongly one response is preferred². From the full preference subset, we use

¹<https://huggingface.co/datasets/nvidia/HelpSteer2>

²`preference_strength` takes the range from -3 to 3, where -3 means response 1 is strongly preferred than response 2.

approximately 7,000 query-level examples, split into 5,602 training examples, 700 validation examples, and 701 test examples. The test set is held out until the final evaluation.

This dataset is a good fit for our setting because it provides both an observed winner–loser relation and a preference-strength signal. The winner–loser relation is used to supervise local geometric edits, while preference strength is used to weakly calibrate the radial gap between critique vectors.

5.2 Generating critique supervision

HelpSteer2 itself provides pairwise preferences, but not the local and global critiques required by our training objective. To construct these additional supervision signals, we prompt a stronger external language model to generate three critiques for each preference pair. The first is a *local critique*, which explains why the losing response is worse than the winning one and therefore describes a local edit from loser to winner. The second and third are *global critiques*, which separately evaluate the winning and losing responses against a broader notion of an latent ideal answer, without revealing which one is the preferred response.

Concretely, for each prompt x with winning response y_w and losing response y_ℓ , we generate $c_{\text{local}}, c_w, c_\ell$, where c_{local} is an edit-style critique of the loser relative to the winner, and c_w and c_ℓ are defect-to-ideal critiques of the winner and loser, respectively. In practice, the prompting template asks the stronger model to identify what is missing, unclear, or incorrect in a response, rather than to rewrite the answer fully from scratch (see Appendix for the detailed prompts).

Using AI-generated critiques is reasonable for two reasons. First, these critiques are generally high quality and provide substantially richer signal than a binary pairwise label. Second, our method does not require critiques to be perfect reconstructions of a human gold answer. The global critiques serve mainly as geometric scaffolding: they only need to be informative enough to shape the latent response space so that better responses lie closer to their critique-implied targets.

5.3 Models and baselines

Our embedding backbone is Qwen3-Embedding-0.6B, a decoder-only Transformer with 28 layers, hidden size 1024, and 16 attention heads. For each input, we use the last-token hidden state as the pooled representation. On top of this backbone, we train a response encoder $e(x, y)$ and a critique encoder $g(x, y, c)$, each implemented as a 3-layer MLP:

$$1024 \rightarrow 512 \rightarrow 512 \rightarrow 1024.$$

We compare three models.

RLHF reward-model baseline. We train a scalar reward model $r(x, y)$ on winner–loser pairs using the frozen Qwen embedding followed by a 3-layer MLP

$$1024 \rightarrow 512 \rightarrow 512 \rightarrow 1,$$

optimized with the standard pairwise logistic loss. This model has 787,969 trainable parameters.

Head-only critique geometry baseline. We freeze the Qwen backbone and train only the response and critique heads. This tests whether the geometric objective is useful even with a fixed backbone. The model has 2,625,536 trainable parameters.

Full critique geometry model with LoRA. We keep the same heads and additionally fine-tune the Qwen backbone with LoRA on the query and value projections in every Transformer layer. We use LoRA rank $r = 8$, $\alpha = 16$, and dropout 0.05. The full model has 3,772,416 trainable parameters.

The baselines answer different questions. The RLHF model learns a scalar reward directly from pairwise rankings, whereas the critique-based models learn a critique-conditioned geometry. At test time, we rank a pair by comparing global-critique norms:

$$\|g(x, y_w, c_w)\| < \|g(x, y_\ell, c_\ell)\|.$$

If this inequality holds, the model predicts that the winner is closer to an improved answer.

5.4 Results

Table 5.4 reports the main test-set ranking results. The standard RLHF baseline achieves an accuracy of 0.6205. Training only the response and critique MLP heads on top of a frozen Qwen backbone already improves this to 0.6419,

Table 1: Test-set ranking accuracy. For the critique-based models, accuracy is the fraction of examples for which the winning response has a shorter inferred global-critique norm than the losing response.

Method	Trainable Parameters	Test Accuracy
RLHF reward model	787,969	0.6205
Head-only critique geometry	2,625,536	0.6419
Full critique geometry + LoRA	3,772,416	0.7389

suggesting that the critique-based geometric objective carries useful signal even without updating the backbone. The full model, which fine-tunes the Qwen backbone with LoRA in addition to the two heads, achieves 0.7389, substantially outperforming both baselines.

These results support the main claim of the project: critique supervision can be converted into a strong geometric signal for response quality. Importantly, this gain is achieved without requiring human-written gold rewrites, and with a supervision format that may be cheaper and more natural to collect than exhaustive pairwise rankings.

6 Analysis

Table 5.4 shows that the learned critique geometry carries substantial ranking information. Intuitively, this happens because the model is trained so that critiques point toward a latent improved answer. For a prompt x and response y , the corrected point

$$e(x, y) + g(x, y, c)$$

is intended to move the response embedding toward a higher-quality region of the latent space. If a response is already good, then the global critique needed to improve it should be relatively small; if a response is poor, the required correction should be larger. This gives a natural ranking signal:

$$r_{\text{geom}}(x, y, c) = -\|g(x, y, c)\|_2^2.$$

A higher value of this score means that the response lies closer to the latent ideal response implied by the critique. Our test-set results indicate that this geometric distance is informative enough to distinguish better from worse responses, even though the model was not explicitly trained as a standard scalar reward model.

More broadly, the value of this method is not exhausted by the ranking numbers in Table 5.4. A standard RLHF reward model only learns an ordering between responses, typically from chosen–rejected pairs. By contrast, our method tries to shape the entire latent response space. The response encoder $e(x, y)$ represents where a response currently lies, while the critique encoder $g(x, y, c)$ represents how that response should move in order to improve. This means the supervision is richer than a binary preference label: a critique provides both directional information, describing *how* the response should change, and magnitude information, describing *how far* the response may be from a better answer. In this sense, the method is not merely another way to produce a scalar reward, but a way to learn structured improvement signals from natural-language feedback.

This distinction matters in practice because critiques are often cheaper and more natural to obtain than the supervision used in standard post-training pipelines. In supervised fine-tuning, human experts are asked to write full target responses, which is expensive and time-consuming. In preference-based RLHF, annotators are asked to compare multiple long responses, which can also be costly and noisy. By contrast, a short critique such as “this answer ignores the causal mechanism” or “the response is correct but too vague” is often much easier to write. Our framework suggests that such critique data can be converted into usable supervision without requiring the annotator to produce the full ideal answer or carefully rank many alternatives.

One practical use case is critique-based post-training from everyday user interactions. Many real deployment settings already generate large amounts of free-form feedback: users say that an answer is unclear, incomplete, overly verbose, or misses the point. Such feedback is usually too unstructured to fit directly into standard RLHF pipelines, but it fits naturally into our framework. If the model can learn a stable geometry in which critiques correspond to improvement directions, then these conversational critiques can become a scalable source of supervision for alignment.

A second use case is reducing the burden of expert annotation in SFT. Instead of asking a domain expert to write a complete ideal response y^* , one could ask for a short critique of an imperfect answer. This is particularly attractive in domains where writing a perfect response is difficult, but identifying what is wrong is much easier. Under our framework, these critiques still provide a meaningful training signal because they define a direction toward improvement even when the exact ideal answer is not written down explicitly.

Finally, the geometric view may be useful beyond ranking itself. Because the learned representation is intended to encode directions of improvement, it may support downstream procedures such as response revision, critique-conditioned reranking, or distillation into a standard critique-free reward model. We do not fully solve these extensions in this project, but the results suggest that critiques contain enough structure to make such directions plausible. This is, in our view, the broader contribution of the work: not simply a better reward score on one benchmark, but a framework for turning natural-language critiques into scalable and structured supervision for language model improvement.

7 Conclusion

This project explores critiques as an alternative supervision format for language model post-training. Instead of relying only on pairwise rankings or fully written target responses, we propose a geometric framework that learns a response encoder $e(x, y)$ and a critique encoder $g(x, y, c)$, where critiques act as vectors pointing toward improved responses. Empirically, the learned geometry provides a strong ranking signal on held-out data and outperforms a simple RLHF reward-model baseline in our experiments.

More importantly, the contribution of this work is not only a new ranking method, but a new way to think about supervision for alignment. Natural-language critiques are abundant, cheaper to provide than full rewrites, and richer than binary preference labels. Our results suggest that such critiques can be converted into structured signals for language model improvement. We hope this project can serve as a step toward more scalable and flexible post-training methods that learn not just which response is better, but how a response should improve.

References

- Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*, 2024.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2024.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *The twelfth international conference on learning representations*, 2023.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Zihuiwen Ye, Fraser David Greenlee, Max Bartolo, Phil Blunsom, Jon Ander Campos, and Matthias Gallé. Improving reward models with synthetic critiques. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4506–4520, 2025.
- Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuwei Wang, Suchin Gururangan, Chao Zhang, et al. Self-generated critiques boost reward modeling for language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11499–11514, 2025.

Appendix: Critique Generation with gemini-2.5-flash

To construct the additional supervision required by our training objective, we use a `gemini-2.5-flash` model to generate critiques for each HelpSteer2 preference pair. For a prompt x with winning response y_w and losing response y_ℓ , we generate three critiques $c_{\text{local}}, c_w, c_\ell$. Here, c_{local} is a local critique of the losing response relative to the winning one, while c_w and c_ℓ are global critiques of the winning and losing responses, respectively. The local critique is designed to capture the main edit needed to move the loser toward the better response. The global critiques are designed to compare each response independently against an ideal answer, without explicitly comparing the two responses to each other.

We use `gemini-2.5-flash` because it provides short, coherent, and actionable critiques at relatively low cost and latency. Importantly, our method does not require these critiques to be perfect gold explanations. The geometric objective only requires critiques to contain useful directional information about improvement, so AI-generated critiques can serve as effective structured supervision even if they are somewhat noisy.

A key design choice is that the model is asked to compare each response against an implicit ideal answer to the user query, rather than against a scoring rubric alone. This framing helps the critiques focus on substantive gaps such as missing information, incorrect emphasis, weak organization, or unnecessary detail. For local critiques, the model is prompted to describe the main ways the weaker response falls short relative to the stronger one, in natural feedback language rather than explicit edit instructions.

For global critiques, we explicitly require each response to be judged independently. The prompt forbids the model from mentioning the other response, leaking comparative information, or revealing which response is better. This is important because the global critique is intended to reflect how far a single response is from an ideal answer, not how it compares to its paired alternative. We also instruct the model not to sound like a rubric or a grader, but instead like a realistic human expert giving brief, direct feedback.

Table 2 and Table 3 show the exact prompt templates used to generate the global and local critiques with `gemini-2.5-flash`.

Table 2: Prompt template for generating global critiques with gemini-2.5-flash.

You are simulating a realistic human user giving brief feedback on two LLM responses.

You are given a user question and two responses. Do not assume either response is better. For each response, compare it independently to what an ideal answer to the question should do.

Write one short critique for Response 1 and one short critique for Response 2.

Requirements:

- Critique each response independently. Do not mention, compare to, or imply the existence of any other response.
- Sound like natural human feedback, not like a formal rubric.
- Mention only the most important one or two issues for each response.
- Be concise, natural, and actionable.
- Do not write the ideal answer.
- Do not compare the two responses explicitly.
- Use plain English.
- Write in third person.
- Maximum 50 words per critique.
- Output exactly in this format:

Critique 1: <critique for Response 1>
Critique 2: <critique for Response 2>

User question:
{question}

Response 1:
{response_1}

Response 2:
{response_2}

Table 3: Prompt template for generating local critiques with gemini-2.5-flash.

You are simulating a realistic human user giving brief feedback on a suboptimal LLM response.

You will be shown:

1. a user question,
2. a weaker response,
3. a stronger response,
4. optional human preference notes explaining why the weaker response was not preferred.

Treat the stronger response as if it were the ideal local reference, but do not mention it explicitly.

Write one short critique of the weaker response only.

Requirements:

- Sound like natural human feedback, not like a formal evaluation rubric.
- Mention only the most important one or two issues.
- Make the critique natural, concise, and actionable.
- If the weaker response is still mostly good, keep the criticism mild.
- Do not compare explicitly to the stronger response.
- Do not mention score, ranking, or preference.
- Do not copy the human notes verbatim.
- Use the human notes only as background context.
- Do not quote large parts of the response.
- Use plain English.
- Write in third person.
- Maximum 50 words.
- Output only the critique text.

User question:

{question}

Weaker response:

{loser}

Stronger response:

{winner}

Optional human preference notes:

Statement: {statement}

Elaboration: {elaboration}
