Learning to Control Summaries with Score Ranking

Hongye Liu
Duke University
hongye.liu@duke.edu

Liang Ding
University of Sydney
liangding.liam@gmail.com

Ricardo Henao
Duke University
ricardo.henao@duke.edu

Abstract

Recent advances in summarization research focus on improving summary quality across multiple criteria, such as completeness, conciseness, and faithfulness, by jointly optimizing these dimensions. However, these efforts largely overlook the challenge of controlling summary generation with respect to individual criteria, especially in the presence of their inherent trade-offs. For example, enhancing conciseness can compromise completeness, and *vice versa*. In this work, we address this gap by proposing a loss function that aligns model outputs with fine-grained, model-based evaluation scores (*e.g.*, from FineSurE), enabling both improvement in summary quality and dimension-specific control. Our approach[1] improves the overall quality of the summaries while maintaining the ability to selectively prioritize one criterion over others. Experiments on three pre-trained models (LLaMA, Qwen, and Mistral) demonstrate that our method achieves performance comparable to state-of-the-art summarizers, while uniquely offering strong controllability over individual quality dimensions.

1 Introduction

Automatic text summarization (Hovy, 2015; Tas and Kiyani, 2007) aims to condense long documents into concise text descriptions that preserve the most important information from the original text, and is widely used in real-world applications (Ji et al., 2026; Xu et al., 2026). Despite huge progress in neural summarization (Cheng and Lapata, 2016; Kryściński et al., 2019), most systems are trained to optimize surrogate objectives, such as model likelihood or n-gram overlap metrics, which correlate poorly with human judgment of semantic fidelity, factual consistency, and content balance (Maynez et al., 2020; Pagnoni et al., 2021). Recent model-based evaluators like FineSurE (Song et al., 2024a) and UniSumEval (Lee et al., 2024) instead extract and align *atomic semantic units* or *keyfacts* between the source and summary, delivering fine-grained *completeness* and *conciseness* scores that correlate better with human assessments (Laban et al., 2023).

Recent advances in summarization research focus on improving summary quality across multiple criteria, including completeness, conciseness, and faithfulness. However, current efforts overlook the challenge of controlling summary generation with respect to individual criteria, such as the trade-off between conciseness and completeness. To address this gap, a loss function is proposed that aligns model outputs with fine-grained, model-based evaluation scores, enabling both improvement in summary quality and dimension-specific control. Experiments on three pre-trained models demonstrate that this method achieves performance comparable to state-of-the-art summarizers, while offering strong controllability over individual quality dimensions.

This work proposes a loss function that enables the improvement of summary quality and dimension-specific control, allowing for the selective prioritization of individual quality dimensions such as completeness and conciseness, while achieving comparable performance to state-of-the-art summarizers.

Recent advances in summarization research focus on improving summary quality, but overlook the challenge of controlling summary generation with respect to individual criteria, such as completeness and conciseness. To address this, a loss function is proposed, enabling both improvement in summary quality and dimension-specific control, with experiments demonstrating comparable performance to state-of-the-art summarizers and strong controllability over individual quality dimensions.

Figure 1: Summaries of the abstract prioritizing completeness ($\text{Com}_{\uparrow}$), conciseness ($\text{Con}_{\uparrow}$) and balance (Bal).

On the optimization side, reinforcement learning (RL) methods, *e.g.*, Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Direct Preference Optimization (DPO) (Rafailov et al., 2024), have been applied to directly optimize non-differentiable quality metrics; however, they suffer from high variance, instability, or prohibitive computational costs (Ahmadian et al., 2024; Liu et al., 2024; Yan et al., 2024). More scalable ranking-based objectives, such as those based on contrastive loss (Liu et al., 2022) and margin ranking, encourage higher-quality candidates to outrank inferior ones (Liu et al., 2023b; Chern et al., 2023). However, they typically optimize a single aggregated score and do not offer mechanisms to steer summaries along distinct dimensions such as summary *completeness* or *conciseness*. In fact, studies on the "alignment tax" have shown that improving one dimension often degrades another (Noukhovitch et al., 2023; Guo et al., 2024). Details of the related work are provided in Appendix A.

[1]The source code and model are available, respectively, at Control-Summaries-with-Ranking and Control-Summaries-LLaMA.

Findings of the Association for Computational Linguistics: ACL 2026, pages 35336–35359

In this work, we propose a framework for *optimizing* neural summarization with explicit control over multiple quality dimensions, with a particular focus on adjusting the trade-off between summary completeness and conciseness. We empirically demonstrate such a trade-off in Appendix F. The key ideas we pursue are: i) *fine-grained ranking* by combining a margin-ranking loss to align the model's log-likelihood with model-based quality scores and a maximum-scoring loss to directly push the top candidate toward higher overall quality; ii) *control-oriented loss* that, given a prompt indicating the desire for a more complete (Com$_\uparrow$), concise (Con$_\uparrow$) or balanced (Bal) summary, adjusts the ratio of completeness to conciseness scores to meet the specified intent; and iii) *unified training objective* that integrates margin ranking, maximum scoring, and control losses, allowing one model to flexibly generate summaries optimized for different trade-offs. Figure 1 shows summary examples from our model using the abstract as a source document.

Technically, we implement our approach by fine-tuning LoRA adapters (Hu et al., 2022) on three open source backbone models, namely LLaMA (Touvron et al., 2023a,b; Dubey et al., 2024), Qwen (Yang et al., 2024) and Mistral (Jiang et al., 2023), and evaluate on both *in-domain* (WikiHow (Koupaee and Wang, 2018), CNN/DM (Nallapati et al., 2016), and DialogSum (Chen et al., 2021)), and *out-of-domain* tasks (OpoSum (Angelidis and Lapata, 2018), MeQSum (Abacha and Demner-Fushman, 2019)). Experiments show that our method: i) significantly improves the Spearman rank correlation between model likelihood and model-based scores; ii) improves the (harmonic) mean of completeness and conciseness for (test) summaries; and iii) delivers strong controllability, producing summaries with high fidelity to the specified dimension while maintaining faithfulness (consistency wrt source). Our contributions can be summarized as follows.

- We introduce a joint ranking and scoring framework that aligns generation likelihoods with fine-grained model-based evaluation metrics, achieving superior ranking performance compared to existing baselines.
- We propose a control-oriented loss that steers summary generation toward completeness, conciseness, or balance based on simple prompts.
- We demonstrate the generality of our approach by fine-tuning multiple language models and evaluating across diverse domains, showing consistent gains in overall quality and controllability.

2 Methodology

Problem Definition Our goal is to generate a summary $Y = (y_1, \ldots, y_N)$ from a source document X by modeling the conditional likelihood $p_\theta(Y|X, Z)$, where Z is a prompt that controls the summarization process. Following the standard autoregressive modeling framework, the generation process is formulated as follows:

$$p_\theta(Y|X, Z) = \prod_{i=1}^{N} p_\theta(y_i|y_{<i}, X, Z)\,, \quad (1)$$

where y_i is the i-th generated token, $y_{<i} = (y_1, \ldots, y_{i-1})$ denotes all previously generated tokens and N is the summary length. When $i = 1$, $y_{<1}$ is the empty sequence, and the first token is generated conditioned only on (X, Z). We seek to generate an optimal summary $\tilde{Y}$ by maximum likelihood (ML), *i.e.*, $\tilde{Y} = \arg\max_Y\ p_\theta(Y|X, Z)$. However, finding the exact ML solution (summary) is intractable due to the exponential size of the output space. In practice, we approximate this by generating a set of candidate summaries using nucleus sampling (Holtzman et al., 2019).

To generate *high-quality* summaries, it is desirable to align $\tilde{Y}$ with a chosen quality score $S_D(\tilde{Y}) = S(\tilde{Y}|X, D)$ that evaluates the summary $\tilde{Y}$ given the document X for some evaluation criterion D. In this work, we focus on summary *completeness* and *conciseness*, as defined in Song et al. (2024a), however, there are several others that can be considered (Lloret et al., 2018). Moreover, for generality purposes, we assume that $S(\cdot)$ is provided by a *black-box model*, *i.e.*, a learned scorer that assigns quality scores but does not expose (propagate) gradients. This assumption promotes scalability and modularity, since the scorer can be replaced or improved independently of the generation model and, in principle, can be applied to any generator without retraining. Previous work has also explored differentiable scoring functions, such as learned reward models for RL (Kaelbling et al., 1996), but these approaches require gradient access and often suffer from stability issues during optimization (Schulman et al., 2017). In the remainder of this work, $S(\cdot)$ is a model-based score.

Our objective is to model p_θ such that the generated summary $\tilde{Y}$ achieves a high overall average quality score $S_{\text{sum}}(\tilde{Y}) = S_{\text{com}}(\tilde{Y}) + S_{\text{con}}(\tilde{Y})$, where $S_{\text{com}}(\tilde{Y})$ and $S_{\text{con}}(\tilde{Y})$ are the completeness

and conciseness scores, respectively. In general, we can define $S_{\text{sum}}(\tilde{Y})$ as the sum or average of a given collection of model-based scores of interest. For example, we could also have considered a faithfulness score. It is worth noting that different quality dimensions often involve trade-offs, *i.e.*, improving conciseness may come at the expense of completeness, and *vice versa*. This is justified because these two scores are inherently at odds: *striving for completeness often leads to longer summaries with more information, which can reduce conciseness, whereas optimizing for conciseness typically requires discarding certain details, potentially harming completeness*. Consequently, it is difficult to produce a single summary $\tilde{Y}$ that maximizes both quality scores simultaneously. This trade-off arises from the limited word "budget" available for a summary, thus compressing content inevitably forces choices about what to include and what to omit, making it challenging to excel on all quality metrics of the summary at once. Thus, it is desirable to equip p_θ with a control mechanism that enables it to generate summaries tailored to different dimensions, *e.g.*, completeness or conciseness. Importantly, Z in (1) now not only guides the summarization process but also controls the specific quality dimension. Hence, another goal is to guide the model p_θ to generate a summary $\tilde{Y}$ prioritizing a specific criterion $D = \{\text{com}, \text{con}\}$, while achieving a high-quality score $S_D(\tilde{Y})$.

Optimizing Completeness and Conciseness Based on the benchmark results using FineSurE reported by Lee et al. (2024), existing models exhibit reasonably strong performance in terms of faithfulness but lag behind in completeness and conciseness. Song et al. (2024b) use Direct Preference Optimization (DPO) (Rafailov et al., 2024) to align large language models with human feedback. However, DPO is computationally expensive, provided that one needs to compute losses from two models (the policy network and the reference network). Moreover, Liu and Henao (2025) show that DPO performs poorly on tasks such as ranking, due to its simplistic pairwise ranking logic. Motivated by these observations, we consider adopting the margin ranking (MR) loss (Liu et al., 2023b; Chern et al., 2023; Liu et al., 2022):

$$\mathcal{L}_{\text{MR}} = \sum_{k=1}^{K} \sum_{j>k}^{K} \max(0, s_j - s_k + \lambda_{jk}) \,, \quad (2)$$

where $\{s_k\}_{k=1}^K$ are the log-likelihood of a collection of K summaries $\{\tilde{Y}_k\}_{k=1}^K$, defined as $s_k = \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(y_i|y_{<i}, X, Z)$, and sorted so that $s_k > s_j$ if $S_{\text{sum}}(\tilde{Y}_k) > S_{\text{sum}}(\tilde{Y}_j)$ for all $k, j \in \{1, \ldots, K\}$. We set the margin $\lambda_{jk} = \lambda \times (j - k)$ for a hyperparameter λ, which is selected through cross-validation in our experiments.

Conceptually, the loss in (2) encourages the model to produce outputs whose log-likelihood scores are ordered consistently with the model-based quality scores $\{S_{\text{sum}}(\tilde{Y}_k)\}_{k=1}^K$, penalizing pairwise order violations. Moreover, the margin enforces a minimum separation between outputs, improving robustness as supported by the literature on maximum-margin methods (Smola, 2000).

An unintended consequence of the loss in (2) is that it focuses only on preserving the order of $\{S_{\text{sum}}(\tilde{Y}_k)\}_{k=1}^K$, without considering their quality. To address this and following Liu and Henao (2025), we introduce an additional objective that directly encourages maximizing the model-based score of the generated summary $\tilde{Y}$ relative to the best reference summary from $\{\tilde{Y}_k\}_{k=1}^K$. We define

$$\mathcal{L}_{\text{MS}} = \max\left(0, \left(S_{\text{sum}}(Y_{\text{ref}}) - S_{\text{sum}}(\tilde{Y})\right) f(s_1)\right) \,, \quad (3)$$

where s_1 is the log-likelihood corresponding to the ML $\tilde{Y}$ under $p_\theta(Y|X, Z)$, $f(\cdot)$ is the exponential function (used to transform log-likelihood to likelihood), and Y_{ref} is the reference summary taken from the set $\{\tilde{Y}_k\}_{k=1}^K$, which is one with the highest model-based score among $S_{\text{sum}}(\{\tilde{Y}_k\}_{k=1}^K)$.

Recall that $\tilde{Y}$ is generated using nucleus sampling. Thus, $\mathcal{L}_{\text{MS}}$ aims to improve the model-based quality score of the top prediction relative to the best reference. As shown in Figure 2, the MR loss $\mathcal{L}_{\text{MR}}$ encourages the alignment of log-likelihoods with model-based scores for a set of K generated summaries, which in the figure is indicated in blue and red for correct and incorrect alignments, respectively. Moreover, the MS loss $\mathcal{L}_{\text{MS}}$ encourages the model to assign higher scores to the top prediction, thereby improving summary quality overall.

So far, the objective in (2) and (3) is to produce high-quality summaries. Next, we propose a learning strategy to train a model capable of generating controllable summaries.

Optimizing Controllability Song et al. (2024b) examine how focusing on a specific dimension affects summary quality compared to considering all dimensions equally. Their experiments shed light on the concept of "alignment tax" (Noukhovitch et al., 2023; Guo et al., 2024), which refers to the trade-off where enhancing alignment with one objective (*e.g.*, conciseness) may reduce performance

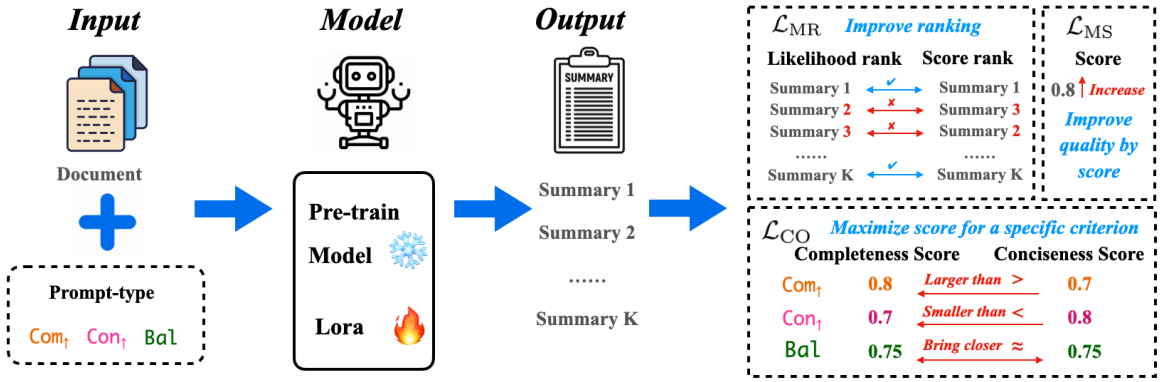

Figure 2: Model architecture and loss functions. The model takes an input document along with a specific prompt to generates K summaries, which are used to compute different loss components. The prompts for Com$_\uparrow$, Con$_\uparrow$ and Bal are designed to prioritize a more complete, more concise or a balanced summary, respectively.

in another (*e.g.*, completeness). However, existing approaches do not effectively control the summary quality along specific dimensions (*e.g.*, completeness or conciseness). To address this limitation, we incorporate the ratio between completeness and conciseness scores as a training signal, guiding the model to shift the summary toward the desired summarization dimensions during training.

For simplicity, we categorize the generation process into three scenarios: i) prioritizing completeness (Com$_\uparrow$), ii) prioritizing conciseness (Con$_\uparrow$), and iii) balancing completeness and conciseness (Bal). The prompt Z in (1) thus not only guides the summary generation, but also specifies the desired quality trade-off. Details of the prompts used for each scenario are provided in Appendix D.

To capture these three scenarios, we first define $S_{\text{ratio}}(\tilde{Y}) = S_{\text{com}}(\tilde{Y})/S_{\text{con}}(\tilde{Y})$. For scenario i), we maximize $S_{\text{ratio}}(\tilde{Y})$; for scenario ii), we maximize $S_{\text{ratio}}(\tilde{Y})^{-1}$; and for scenario iii), we aim for $S_{\text{ratio}}(\tilde{Y}) \to 1$. We define the control-oriented loss as follows.

$$\mathcal{L}_{\text{CO}} = \begin{cases} \max(0,\, [S_{\text{ratio}}(Y_{\text{ref}}) - S_{\text{ratio}}(\tilde{Y})] f(s_1)), & \text{Com}_\uparrow, \\ \max(0,\, [S_{\text{ratio}}(\tilde{Y}) - S_{\text{ratio}}(Y_{\text{ref}})] f(s_1)), & \text{Con}_\uparrow, \\ \max(0,\, [|\log S_{\text{ratio}}(\tilde{Y})| - |\log S_{\text{ratio}}(Y_{\text{ref}})|] f(s_1)), & \text{Bal}, \end{cases} \quad (4)$$

where s_1 is the log-likelihood of the ML generated summary $\tilde{Y}$, $f(\cdot)$ is the exponential function (used to transform log-likelihood to likelihood), and Y_{ref} is the reference summary selected from the set $\{\tilde{Y}_k\}_{k=1}^K$, which is the one with the highest model-based score among $S_{\text{ratio}}(\{\tilde{Y}_k\}_{k=1}^K)$. This loss enforces the desired adjustment in the ratio $S_{\text{ratio}}(\tilde{Y})$, pushing it higher when completeness is prioritized, lower when conciseness is prioritized, and closer to 1 in the balanced case.

As shown in Figure 2, the control loss $\mathcal{L}_{\text{CO}}$ introduces prompt-specific control signals. Input prompts are categorized into one of three types: Com$_\uparrow$, Con$_\uparrow$ or Bal. The model learns to condition its output on these control types, favoring completeness over conciseness for the first, conciseness over completeness for the second, and balancing both for the third. This enables fine-grained control over summary characteristics, improving flexibility and alignment with user intent.

Although $\mathcal{L}_{\text{CO}}$ enables the model to generate summaries aligned with prompt-specific preferences, the scoring mechanism in (2) applies a uniform aggregation of completeness and conciseness scores, which lacks the granularity needed to reflect these nuanced control signals. In (2), the summaries are ranked according to their overall score $S_{\text{sum}}(\tilde{Y}) = S_{\text{com}}(\tilde{Y}) + S_{\text{con}}(\tilde{Y})$. However, this simple aggregation fails to reflect fine-grained preferences. For example, summaries with $(S_{\text{com}} = 0.9, S_{\text{con}} = 0.1)$, $(S_{\text{com}} = 0.1, S_{\text{con}} = 0.9)$, and $(S_{\text{com}} = S_{\text{con}} = 0.5)$ produce the same S_{sum}, but exhibit very different trade-offs. So motivated, we update $S_{\text{sum}} \to S_{\text{sum}*}$ in (2) to incorporate scenario-specific penalties. Specifically,

$$S_{\text{sum}*} = S_{\text{sum}} - \delta \cdot \phi(S_{\text{com}}, S_{\text{con}}), \quad (5)$$

with the penalty term $\phi(\cdot, \cdot)$ defined as:

$$\phi(S_{\text{com}}, S_{\text{con}}) = \begin{cases} S_{\text{con}} - S_{\text{com}}, & \text{Com}_\uparrow, \\ S_{\text{com}} - S_{\text{con}}, & \text{Con}_\uparrow, \\ |S_{\text{com}} - S_{\text{con}}|, & \text{Bal}, \end{cases}$$

where $\delta > 0$ is a hyperparameter (set using cross-validation) that determines the strength of the penalty for deviation from the desired trade-off. This loss encourages fine-grained priorities to better align with the specified criteria when the average scores of completeness and conciseness are identical across options. By subtracting $\delta \cdot \phi$ from

S_{sum}, $S_{\text{sum}*}$ becomes a more discriminative scoring function that aligns better with task-specific preferences, even when raw scores S_{sum} are tied.

Finally, we combine the margin ranking loss in (2) modified using (5) with the score-improving loss in (3) and control-oriented loss (4) as

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{MR}} + \gamma \mathcal{L}_{\text{MS}} + \beta \mathcal{L}_{\text{CO}}, \quad (6)$$

where γ is a hyperparameter that balances the improvement of model-based scores and β balances controllability. These $\{\gamma, \beta\}$ are tuned using cross-validation in our experiments. Details of the FineSurE calculation and the corresponding prompt designs can be found in Appendices B and C.

3 Experiments

Datasets We consider three datasets from five different domains. The FeedSum dataset (Song et al., 2024b) includes separate training and test sets drawn from different domains of documents. The training set contains a diverse collection of model-generated summaries, each annotated with fine-grained FineSurE scores. Due to computational constraints, we focus on short document types in this study. Specifically, we use three in-domain datasets for training and evaluation, namely Wikihow (lifestyle) (Koupaee and Wang, 2018), CNN/DM (news) (Nallapati et al., 2016), and DialogSum (daily life conversations) (Chen et al., 2021). We also evaluate generalization to two out-of-domain summarization datasets: OpoSum (product reviews) (Angelidis and Lapata, 2018) and MeQSum (medical) (Abacha and Demner-Fushman, 2019). In the Appendix, we show Table 7, which summarizes these three datasets.

Baselines We compare our method with five popular prompt-based LLMs: LLaMA (Touvron et al., 2023a,b; Dubey et al., 2024), Qwen (Yang et al., 2024), Mistral (Jiang et al., 2023), GPT (Ouyang et al., 2022) and Gemini (Team et al., 2023). For LLaMA, we consider two variants: `LLaMA-3.1-8B-Instruct` and `SummLLaMA3-8B`, the latter being a fine-tuned version trained on the FeedSum dataset (Song et al., 2024b). For Qwen, we use the `Qwen2.5-7B-Instruct` model, and for Mistral, we evaluate `Mistral-7B-Instruct`. For GPT, we consider `GPT-4o`, `GPT-4o-mini`, and `GPT-4-turbo`. For Gemini, we only use `gemini-2.0-flash`. The prompts for controlled summary generation can be found in Appendix D.

Evaluation Metrics We use Spearman's rank correlation coefficient to measure the alignment between model predictions $\log p_\theta(\{\tilde{Y}_k\}_{k=1}^K$ and model-based scores $S_{\text{sum}}(\{\tilde{Y}_k\}_{k=1}^K)$. A higher Spearman correlation indicates stronger alignment between the model's likelihood estimates and the model-based scores, suggesting that the model assigns higher probabilities to summaries that are rated more favorably. We also consider the harmonic mean of completeness and conciseness scores to evaluate the overall quality of a summary: $\text{HM}(\tilde{Y}) = 2/(S_{\text{com}}(\tilde{Y})^{-1} + S_{\text{con}}(\tilde{Y})^{-1})$. A higher harmonic mean $\text{HM}(\tilde{Y})$ is associated with better overall summary quality. Lastly, we obtain the control ratio to assess the ability of the model to guide summaries toward the desired dimension: $\text{R}(\tilde{Y}) = (\log(S_{\text{com}}(\tilde{Y})/S_{\text{con}}(\tilde{Y})))^\alpha$, where $\alpha = 1$ when prioritizing completeness or balance; and $\alpha = -1$ when prioritizing conciseness. We expect $\text{R}(\tilde{Y})$ to be high when completeness or conciseness is explicitly prioritized, depending on the value of α, and to be close to zero when balance is desired.

Implementation Our methodology considers `Qwen2.5-7B-Instruct`, `Mistral-7B-Instruct`, and `LLaMA-3.1-8B-Instruct` using a single NVIDIA H100 GPU. To increase the diversity of the candidate pool, we obtain K-candidate summaries by combining $K-1$ reference summaries randomly sampled from the FeedSum dataset (Song et al., 2024b), which contains summaries generated by various models, with the top summary prediction produced by our model via nucleus sampling. These summaries constitute the set of candidates used to compute the loss. The model is fine-tuned for three epochs, and during each iteration, the newly generated summary is added to the reference summary pool. All models were initialized from pre-trained backbones and fine-tuned using LoRA (Hu et al., 2022). The hyperparameter selection and their settings are in Appendix E.

For ranking-based evaluation, we set our own test split from the FeedSum training data since the original FeedSum (Song et al., 2024b) test set does not contain diverse model-generated summaries with corresponding FineSurE scores. Specifically, we sampled 100 summaries from each domain to form a test set for ranking-based evaluation, and used the remaining 12,000 examples as the training set. The original FeedSum test set was retrained for summary quality evaluation using $\text{HM}(\tilde{Y})$, and control ability using $\text{R}(\tilde{Y})$ and FineSurE scores obtained through GPT-4o, which was selected to align with SummLLaMA (Song et al., 2024b), and thus ensuring a consistent comparison.

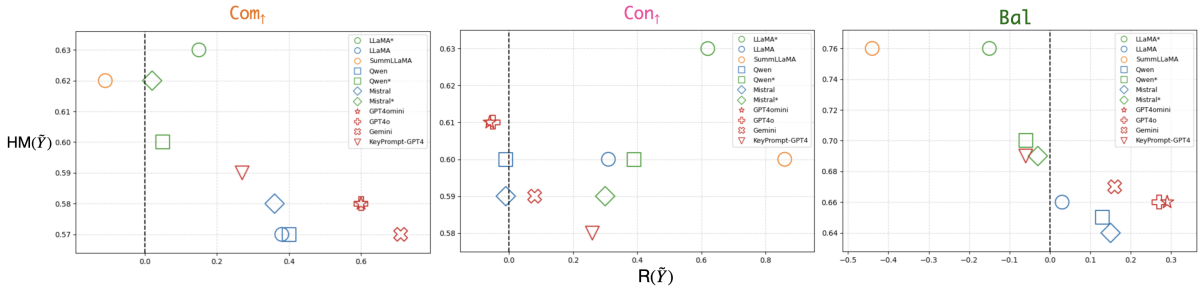

Figure 3: Model performance across different control settings. Each point represents the mean of HM($\tilde{Y}$) (y-axis) and R($\tilde{Y}$) (x-axis) across all test cases in FeedSum test set. Models are grouped by color into four categories: Baseline models (blue), Our methods (green), SummLLaMA (orange), and Commercial models (red). The three panels show performance under different control priorities: Com$_\uparrow$ prioritizes completeness, Con$_\uparrow$ prioritizes conciseness, and Bal aims to balance both. The vertical dashed line at R($\tilde{Y}$) = 0 represents the controllability target (reference) between completeness and conciseness.

Model	Median	IQR
LLaMA	0	(-0.3, 0.22)
SummLLaMA	0.04	(-0.26, 0.38)
LLaMA*	0.15	(-0.19, 0.37)
Qwen	-0.02	(-0.34, 0.29)
Qwen*	**0.22**	(-0.02, 0.46)
Mistral	-0.03	(-0.23, 0.19)
Mistral*	0.09	(-0.19, 0.38)

Table 1: Spearman correlation between model likelihood and model-based scores for K summaries and all cases in the test dataset. We report the median and interquartile range (IQR). The best result is highlighted in **bold**. The asterisk (*) indicates the base foundation model fine-tuned using the proposed method.

Ranking performance Section 2 introduced our approach to aligning model predictions with model-based scores, specifically using the margin-ranking objective ($\mathcal{L}_{\text{MR}}$) in (2). Table 1 shows the results using (median) Spearman correlations to compare alignment quality. These are reported for all cases in the FeedSum test dataset. For completeness, we also report the Pearson and Kendall's tau correlation metrics in the Appendix Table 6.

Among all models evaluated, Qwen* achieves the highest median Spearman correlation (0.22), with an interquartile range of (-0.02, 0.46). Qwen* refers to the Qwen base model fine-tuned using the proposed method. This result suggests that our alignment approach is effective when applied to the Qwen architecture. LLaMA*, ranks second with a median score of 0.15 and a relatively tight IQR of (-0.19, 0.37), which outperforms its base version (LLaMA, median 0.00) and the variant SummLLaMA (median 0.04), further demonstrating the general effectiveness of our training strategy in enhancing alignment with quality scores. The consistently higher Spearman correlations for all our fine-tuned models indicate that our method may also be useful for summary quality ranking via model likelihoods. The Spearman scores distributions for all models are shown in Appendix Figure 5.

Quality and Controllability Performance In (3) ($\mathcal{L}_{\text{MS}}$), we introduced a loss to improve the quality of model-generated summaries. We now evaluate the quality of these summaries using FineSurE. Specifically, we seek high completeness *and* conciseness scores for generated summaries, which we evaluate using the HM($\tilde{Y}$) score introduced above. Complementary, for controllability performance we seek high R($\tilde{Y}$) values for Com$_\uparrow$ and Con$_\uparrow$, which imply that the model assigns greater importance to the higher-priority attribute, completeness or conciseness, respectively; or R($\tilde{Y}$) $\rightarrow 0$ when a balanced (Bal) summary is of interest. In general, the objective is to maintain high quality generation while ensuring effective control. Therefore, a higher HM($\tilde{Y}$) value is preferred, along with preferably a higher R($\tilde{Y}$).

Figure 3 separately shows the mean HM($\tilde{Y}$) and R($\tilde{Y}$) metrics for the three control scenarios (Com$_\uparrow$, Con$_\uparrow$ and Bal), 10 different methods (3 baseline open source models, 3 corresponding fine-tuned models, SummLLaMa and three commercial models), for all the test cases in the FeedSum test set. We observe the following: i) In the (Com$_\uparrow$) and (Con$_\uparrow$) scenarios, we seek models on the right-hand side of the R($\tilde{Y}$) scale (*i.e.*, with R($\tilde{Y}$) > 0) while also producing high HM($\tilde{Y}$) values. However, SummLLaMA (orange) achieves reasonable HM($\tilde{Y}$) scores but consistently produces negative R($\tilde{Y}$) values in the (Com$_\uparrow$) setting, favoring conciseness regardless of the intended control. This indicates limited controllability, which is likely due to its optimization for general summarization quality rather than controllable generation. ii) Our models (green), especially LLaMA*, achieve high HM($\tilde{Y}$) scores and positive R($\tilde{Y}$) values, demonstrating effective controllability and strong summary quality. iii) In the (Bal) setting, we aim for models near the dashed line (R($\tilde{Y}$)$\rightarrow$ 0), in-

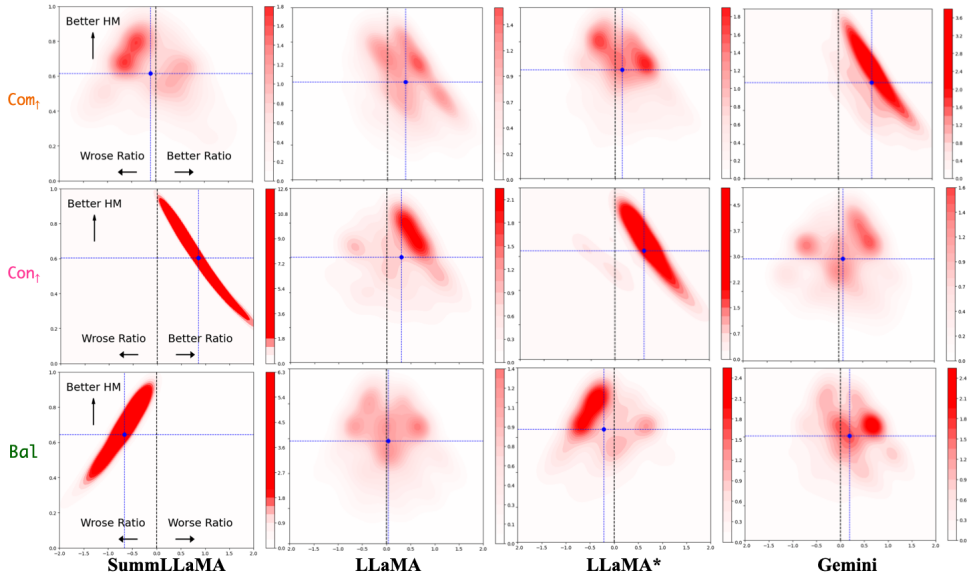

Figure 4: Distributions of R($\tilde{Y}$) (x) and HM($\tilde{Y}$) (y) metrics. Com$_\uparrow$ prioritizes completeness, Con$_\uparrow$ prioritizes conciseness, and Bal to balance them. Blue dashed lines mark the mean of the metrics, and the arrows point in the direction in which metrics are better or worse.

dicating balanced attention to both completeness and conciseness, together with high HM($\tilde{Y}$). Here, LLaMA* matches SummLLaMA in HM($\tilde{Y}$) but is much closer to the target R($\tilde{Y}$), showing that our method achieves better controllability without sacrificing quality. iv) Baseline models (blue) generally exhibit higher controllability, but underperform in terms of HM($\tilde{Y}$) compared to our methods. v) Commercial models (red) exhibit good controllability (R($\tilde{Y}$) aligned with the desired direction), but their overall HM($\tilde{Y}$) scores are lower. vi) Keyprompt is a prompt-oriented baseline in which we prompt the model to first generate key facts and then produce a summary conditioned on these key facts. Although the prompt-based method offers flexibility and achieves reasonable controllability (*i.e.*, R($\tilde{Y}$)), it comes at a significant cost in terms of overall summary quality, with its HM($\tilde{Y}$) being considerably lower than our fine-tuned LLaMA* across all control settings.

These results demonstrate that our models consistently achieve strong performance and robust control, outperforming both baselines and commercial systems in balancing quality with controllability. Detailed values for all baseline model comparisons are provided separately in Appendix Table 10.

To better illustrate the ability of each model to generate high-quality summaries (by HM($\tilde{Y}$)) and controllability (by R($\tilde{Y}$)), we show their distributions (as contours) in Figure 4, where the x and y axes represent R($\tilde{Y}$) and HM($\tilde{Y}$), respectively, and the color indicates density. The blue marker is the mean HM($\tilde{Y}$) and R($\tilde{Y}$) from Table 10. We expect distributions to be concentrated in the upper-middle region, reflecting both high summary quality and effective control. For the scenarios Com$_\uparrow$ and Con$_\uparrow$, we hope to see the distribution tail extend to the right, which indicates a consistent control bias aligned with the prompt while maintaining high quality. Under the Bal setting, we hope for a clustered distribution in the upper-middle region, thus summaries that are both complete and concise.

As shown in Figure 4, among all models: i) LLaMA* (our method) consistently produces the most favorable distribution, concentrated in the desired region for all three settings, and generally shifted upward (indicating high HM($\tilde{Y}$)), suggesting both strong control and high-quality output. ii) In contrast, across all settings, SummLLaMA consistently produces overly concise summaries (evidenced by a leftward shift in the Com$_\uparrow$ and Bal settings), which indicates poor controllability. iii) LLaMA demonstrates better control alignment (*i.e.*, correct shift in R($\tilde{Y}$)), but its lower HM($\tilde{Y}$) values suggest limited summary quality. Results for all models are shown in Appendix Figures 7 and 8.

Ablation study We conducted an ablation study to assess the individual contributions of each component in our loss function: MR (ranking), MS (generation quality), and CO (control), all under the same training setup.

In Table 2, we compute the Spearman correlation between model likelihoods and model-based evaluation scores over K sampled summaries for all test cases, under different LLaMA fine-tuning settings with ablated loss components. This eval-

Model	Median	IQR
LLaMA* (MR)	0.07	$(-0.27, 0.37)$
LLaMA* (MS)	0.04	$(-0.26, 0.38)$
LLaMA* (CO)	0.05	$(-0.26, 0.38)$
LLaMA* (MR+MS)	0.14	$(-0.16, 0.38)$
LLaMA* (MR+CO)	0.12	$(-0.16, 0.36)$
LLaMA* (MS+CO)	0.04	$(-0.26, 0.38)$
LLaMA* (MR+MS+CO)	**0.15**	$(-0.19, 0.37)$

Table 2: Spearman correlation analysis for ablation on LLaMA loss components. The best results are highlighted in bold, and the asterisk ($*$) denotes the base LLaMA model fine-tuned using the proposed objective.

uates how well the model aligns with summary quality. We report the median and interquartile range (IQR) for each configuration. Using MR alone yields only limited improvements in ranking performance. However, when MR is combined with MS or CO, the ranking results improve noticeably. In fact, the full combination of MR, MS, and CO achieves the highest median correlation, indicating a strong synergy among the losses. For control quality, in Appendix Table 8, we show that incorporating MS leads to better generation quality, reflected in higher HM($\tilde{Y}$) scores, while adding CO enhances controllability, as shown by improvements in R($\tilde{Y}$). The combination of the three losses results in a well-balanced trade-off between summary quality and controllability. Thus, these findings highlight the complementary roles of the loss components and provide empirical support for the effectiveness of our composite training objective.

Human Study To assess whether our model effectively learns to generate summaries with controllable attributes, we conducted a human evaluation based on pairwise comparisons.

We use our fine-tuned model (LLaMA*) to generate summaries under different control prompts (Com$_\uparrow$, Bal, Con$_\uparrow$) and construct pairwise comparisons between them. To reduce potential biases related to length, we filtered for summary pairs in which both outputs contained the same number of sentences. From this filtered set, we sampled 60 summary pairs for human annotation. These were divided into five sets of annotations, each containing 20 pairs. The first 10 pairs were shared among all annotators to measure inter-annotator agreement (IAA), while the remaining 10 were unique to each annotator. The evaluation of the human-model alignment used all pairs. For this task, annotators were asked to read two summaries generated from the same document and select the one they considered more complete. No additional instructions on conciseness or fluency were provided, as the focus was solely on completeness. Then, their preferences were compared to the control signal given to the model. We assumed an ordinal relationship among the control settings (Com$_\uparrow$ > Bal > Con$_\uparrow$), and based on this, assigned a pseudo-label to each pair, *e.g.*, if summary A was generated under a higher completeness setting than summary B, it was labeled 1, indicating that A should be more complete; otherwise, labeled 0.

	A_1	A_2	A_3	A_4	A_5
A_1	1.00	0.60	0.80	0.60	1.00
A_2	-	1.00	0.40	0.60	0.60
A_3	-	-	1.00	0.40	0.80
A_4	-	-	-	1.00	0.60
A_5	-	-	-	-	1.00
Average $\pm$ Std			0.64 ± 0.18		

Table 3: Inter-annotator agreement matrix for Cohen's kappa (κ) with average and standard deviation.

Annotator	Accuracy	Spearman (ρ)
A_1	0.95	0.903
A_2	0.80	0.612
A_3	0.65	0.385
A_4	0.95	0.903
A_5	0.90	0.816
Average $\pm$ Std	$\mathbf{0.85 \pm 0.18}$	$\mathbf{0.72 \pm 0.22}$

Table 4: Annotator performance in terms of Accuracy and Spearman correlation (ρ).

Five annotators participated in the study. IAA was assessed using Cohen's kappa (κ) on the 10 shared examples. Table 3 shows an average $\kappa = 0.64 \pm 0.18$, indicating moderate to substantial agreement and consistent annotation behavior among annotators. To evaluate model-human alignment, we computed the accuracy and Spearman's rank correlation (ρ) between the model's pseudo-labels and human preferences, using all 20 examples per annotator. Table 4 reports an average accuracy of 0.85 ± 0.18 and an average Spearman's of $\rho = 0.724 \pm 0.22$. These results demonstrate a strong alignment between the model's control-guided outputs and human judgments, supporting the effectiveness of our approach for controllable summary generation.

Domain-Specific Results To assess domain adaptation capabilities, we analyze the performance on WikiHow (lifestyle), CNN/DM (news), and DialogSum (dialog), as shown in Appendix Figure 10. We observe the following: i) Our model achieves strong HM($\tilde{Y}$) and R($\tilde{Y}$) scores in all domains. However, in the (Com$\uparrow$) setting on CNN/DM, models such as Qwen* and Mistral* produce negative R($\tilde{Y}$), indicating the difficulty in generating complete summaries. ii) In contrast, commercial mod-

els consistently achieve positive $R(\tilde{Y})$ under the (Com↑) setting across all domains, demonstrating their effectiveness in prioritizing completeness. *iii*) SummLLaMA consistently exhibits negative $R(\tilde{Y})$ in both ($Com_{\uparrow}$) and (Bal) settings, despite achieving high $HM(\tilde{Y})$ scores, highlighting a lack of controllability across domains. To evaluate out-of-domain robustness, we assess performance in OpoSum (product reviews) and MeQSum (medical summaries), as shown in Appendix Figure 9. Similar trends are observed: *i*) Commercial models lead in both $HM(\tilde{Y})$ and $R(\tilde{Y})$, indicating that open-source models still have room for improvement. *ii*) our method outperforms the baselines in both $HM(\tilde{Y})$ and controllability, and SummLLaMA remains overly concise. The complete results are provided in Appendix Tables 14 and 13.

Extended Quality Evaluation To prevent *reward hacking* (Skalse et al., 2022), we evaluate models using G-Eval+ (Lee et al., 2024), as shown in Appendix Figure 6. We observe the following: *i*) Nearly all models exhibit negative $R(\tilde{Y})$ values under the $Com_{\uparrow}$ and Bal settings, indicating limited controllability. Our method mitigates this issue under the LLaMA and Qwen frameworks, shifting the $R(\tilde{Y})$ toward the positive direction while also improving $HM(\tilde{Y})$. *ii*) LLaMA* consistently outperforms both LLaMA and SummLLaMA in terms of overall quality ($HM(\tilde{Y})$) and controllability ($R(\tilde{Y})$), demonstrating a better balance between the two. *iii*) Commercial models achieve strong performance, with consistently high scores in both $HM(\tilde{Y})$ and $R(\tilde{Y})$, reflecting superior summary quality and controllability. The complete results are in Appendix Table 12. Moreover, we considered additional dimensions (consistency, coherence, fluency and relevance) using G-Eval (Liu et al., 2023a), and ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2019) compared against human reference summaries. The results in Appendix Table 9 show that our model (LLaMA*) consistently outperforms the SummLLaMA baseline in most G-Eval metrics and shows clear improvements over untrained LLaMA, demonstrating the effectiveness of our control-aware training. Improvements in ROUGE and BERTScore are comparatively smaller, likely due to a broad overlap with the reference summaries. We also report the faithfulness score in Table 11 and Table 12, showing that our method maintains high faithfulness.

Dimension Extension To verify the effectiveness of our method, we also conduct experiments along different dimensions. In this setting, we focus on completeness and faithfulness. As shown in Appendix Figures 11 and 12, both the baseline model and SummLLaMA struggle to achieve positive performance in the $Com_{\uparrow}$ setting ($R(\tilde{Y}) > 0$). In contrast, after training with our approach, the model can be effectively guided along the desired dimension. We report the detailed results in Table 15.

4 Discussion

Our proposed method is related to Reinforcement Learning from AI Feedback (RLAIF) in that it learns from AI feedback. However, the key distinction lies in the training objective and computational requirements. Standard RL approaches, such as PPO (Schulman et al., 2017) or GRPO (Shao et al., 2024), require keeping a reference model to compute the policy loss and to constrain policy updates. This design increases both memory consumption and training cost; as it involves additional forward passes and parameter storage. In contrast, our method adopts a ranking-based objective rather than a reinforcement learning objective. It does not require hosting with a reference model. As a result, the training process is simpler and computationally more efficient. The current work focuses on two important summarization dimensions, while the proposed framework is readily compatible with other perspectives, such as stylistic attributes, which can be incorporated in future work through the same scoring and ranking formulation.

Limitations

The method is heavily dependent on the quality of the external scoring function FineSurE. Although it correlates with human judgments, it remains an imperfect proxy, and thus it can introduce biases or systematic errors. If the scorer fails to capture certain aspects of quality (*e.g.*, fluency or discourse coherence), the model may overfit to these incomplete signals, leading to reward misalignment or reward hacking. The approach also relies on sampling multiple candidate summaries K to compute ranking losses, which introduces additional computational overhead compared to standard maximum likelihood training. Moreover, the training signal is sensitive to the quality and diversity of the sampled candidates. Finally, the framework is validated only on two quality dimensions (*e.g.*, completeness and conciseness). Although the formulation is general, its effectiveness in scenarios involving three or more dimensions remains unexplored.

References

Asma Ben Abacha and Dina Demner-Fushman. 2019. On the summarization of consumer health questions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2228–2234.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. *arXiv preprint arXiv:1808.08858*.

Manik Bhandari, Pranav Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. *arXiv preprint arXiv:2010.07100*.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. Dialogsum: A real-life scenario dialogue summarization dataset. *arXiv preprint arXiv:2105.06762*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494.

I-Chun Chern, Zhiruo Wang, Sanjan Das, Bhavuk Sharma, Pengfei Liu, Graham Neubig, and 1 others. 2023. Improving factuality of abstractive summarization via contrastive reward learning. *arXiv preprint arXiv:2307.04507*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Zexu Sun, Bowen Sun, Huimin Chen, Ruobing Xie, Jie Zhou, Yankai Lin, and 1 others. 2024. Controllable preference optimization: Toward controllable multi-objective alignment. *arXiv preprint arXiv:2402.19085*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Eduard Hovy. 2015. Text summarization. In *The Oxford Handbook of Computational Linguistics*, pages 972–990. Oxford University Press.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Hongru Ji, Yuyin Fan, Meng Zhao, Xianghua Li, Lianwei Wu, and Chao Gao. 2026. Stride-ed: A strategy-grounded stepwise reasoning framework for empathetic dialogue systems. *Preprint*, arXiv:2604.07100.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551.

Philippe Laban, Wojciech Kryściński, Divyansh Agarwal, Alexander Richard Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. Summedits: Measuring llm ability at factual reasoning through the lens of summarization. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9662–9676.

Yuho Lee, Taewon Yun, Jason Cai, Hang Su, and Hwanjun Song. 2024. Unisumeval: Towards unified, fine-grained, multi-dimensional summarization evaluation for llms. *arXiv preprint arXiv:2409.19898*.

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

Hongye Liu and Ricardo Henao. 2025. Learning to substitute words with model-based score ranking. 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 11551–11565.

Wei Liu, Yang Bai, Chengcheng Han, Rongxiang Weng, Jun Xu, Xuezhi Cao, Jingang Wang, and Xunliang Cai. 2024. Length desensitization in directed preference optimization. *arXiv preprint arXiv:2409.06411*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization. *arXiv preprint arXiv:2203.16804*.

Yixin Liu, Kejian Shi, Katherine S He, Longtian Ye, Alexander R Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. 2023b. On learning to summarize with large language models as references. *arXiv preprint arXiv:2305.14239*.

Elena Lloret, Laura Plaza, and Ahmet Aker. 2018. The challenging task of summary evaluation: an overview. *Language Resources and Evaluation*, 52:101–148.

Qingyu Lu, Liang Ding, Liping Xie, Kanjian Zhang, Derek F Wong, and Dacheng Tao. 2023. Toward human-like evaluation for natural language generation with error analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5892–5907.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, and 1 others. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Michael Noukhovitch, Samuel Lavoie, Florian Strub, and Aaron C Courville. 2023. Language model alignment with elastic reset. *Advances in Neural Information Processing Systems*, 36:3439–3461.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471.

Alexander J Smola. 2000. *Advances in large margin classifiers*. MIT press.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024a. Finesure: Fine-grained summarization evaluation using llms. *arXiv preprint arXiv:2407.00908*.

Hwanjun Song, Taewon Yun, Yuho Lee, Jihwan Oh, Gihun Lee, Jason Cai, and Hang Su. 2024b. Learning to summarize from llm-generated feedback. *arXiv preprint arXiv:2410.13116*.

Oguzhan Tas and Farzad Kiyani. 2007. A survey automatic text summarization. *PressAcademia Procedia*, 5(1):205–213.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Chengbing Wang, Yang Zhang, Wenjie Wang, Xiaoyan Zhao, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2025. Think-while-generating: On-the-fly reasoning for personalized long-form generation. *arXiv preprint arXiv:2512.06690*.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. Self-preference bias in llm-as-a-judge. *arXiv preprint arXiv:2410.21819*.

Jiawen Xie, Shaoting Zhang, and Xiaofan Zhang. 2024. Gecsum: Generative evaluation-driven sequence level contrastive learning for abstractive summarization. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7581–7595.

Shijia Xu, Yu Wang, Xiaolong Jia, Zhou Wu, Kai Liu, and April Xiaowen Dong. 2026. RCBSF: A multi-agent framework for automated contract revision via stackelberg game. *Preprint*, arXiv:2604.10740.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Yang Wang. 2024. Pride and prejudice: Llm amplifies self-bias in self-refinement. *arXiv preprint arXiv:2402.11436*.

Yuzi Yan, Xingzhou Lou, Jialian Li, Yiping Zhang, Jian Xie, Chao Yu, Yu Wang, Dong Yan, and Yuan Shen. 2024. Reward-robust rlhf in llms. *arXiv preprint arXiv:2409.15360*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Taewon Yun, Jihwan Oh, Hyangsuk Min, Yuho Lee, Jihwan Bang, Jason Cai, and Hwanjun Song. 2025. Refeed: Multi-dimensional summarization refinement with reflective reasoning on feedback. *arXiv preprint arXiv:2503.21332*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Related Work

Summary Evaluation Traditional summarization metrics (*e.g.*, ROUGE) correlate poorly with human judgments of semantic fidelity and factuality. Recent LLM-based evaluators instead assess content at a finer granularity (Laban et al., 2023; Lu et al., 2023). To be specific, FineSurE (Song et al., 2024a) extracts atomic *keyfacts* from the source and summary, computing completeness and conciseness by aligning these keyfacts with summary sentences, delivering interpretable and dimension-specific scores. UniSumEval (Lee et al., 2024) provides a unified and robust resource for benchmarking summarization systems, facilitating a more accurate and comprehensive performance evaluation. Collectively, these works mark a shift from token-level to content-level evaluation, paving the way for more reliable and actionable supervision. Our work moves a step further by using these scores as a reward signal for model refinement.

Summary Optimization Training summarization models directly with non-differentiable metrics has been attempted via reinforcement learning (RL) and preference learning. Early RL methods treat evaluation scores as rewards (Kaelbling et al., 1996), *e.g.*, Proximal Policy Optimization (PPO) (Schulman et al., 2017) or Direct Preference Optimization (DPO) (Rafailov et al., 2024). However, they come at the cost of either high variance and instability or substantial computation cost.

Ranking-based objectives offer a more scalable alternative. BRIO (Liu et al., 2022) casts summarization as a contrastive ranking task, encouraging higher-quality candidates to outrank inferior ones. Subsequent work introduces margin ranking losses to enforce explicit quality separations (Liu et al., 2023b; Chern et al., 2023). Contrastive sequence learning approaches like GEC-Sum (Xie et al., 2024) further integrate automated evaluation scores into end-to-end fine-tuning. More recently, Song et al. (2024b) showed that LLM-generated critiques can serve as effective supervision signals, closing the gap between human feedback and model training. Yun et al. (2025) propose an inference-time refinement method that performs multiple iterations for a single input. Given a document, the model first generates an initial summary and then computes its FineSurE score. This score is fed back to the model as a feedback signal to refine the summary, and the process is repeated until the summary reaches a sufficiently high score.

Despite these advances, most methods optimize a single aggregated score and lack mechanisms to steer generation along distinct quality dimensions. Importantly, controlling or steering model generation has been increasingly recognized as essential (Wang et al., 2025). This is further supported by work on the "alignment tax" (Noukhovitch et al., 2023; Guo et al., 2024), which reveals that improving one objective (*e.g.*, conciseness) often degrades another (*e.g.*, completeness). In contrast, our approach combines fine-grained LLM-based scoring, margin ranking, and a control-oriented loss to enable high-quality summaries that can be flexibly tuned toward completeness, conciseness, or a balanced trade-off.

B FineSurE

The Finegrained Summarization Evaluation (FineSurE) model is a recently proposed model-based score that uses large language models (LLMs) to evaluate the summarization quality of a generator at a fine-grained level through summary sentences they call *keyfacts* (Song et al., 2024a). A keyfact is defined as a short sentence conveying a single key information element, consisting of at most 2 or 3 entities, where an entity refers to a salient concept, object, or named item (*e.g.*, people, organizations, locations, or events). These keyfacts are also known as *semantic content units* (Bhandari et al., 2020).

To measure the completeness and conciseness of a generated summary, we rely on a process called *keyfact alignment*, which assesses how well the content of the summary corresponds to the key information elements in the source document. The alignment of keyfacts serves as the foundation for computing both completeness and conciseness scores. It involves determining which keyfacts extracted from the source document are present in the summary and then identifying the summary sentences that express them. Although humans are generally best at generating keyfacts, particularly in specialized domains such as medicine or sales, where content prioritization requires domain expertise, obtaining such annotations can be costly or impractical. FineSurE is designed to use human-provided keyfacts when available. Alternatively, both keyfact extraction and alignment are automatically performed using the LLM with task-specific prompts, as shown in Appendix C.

Suppose that after keyfact extraction, a docu-

ment X contains $\hat{n}$ keyfacts and the summary Y contains $\tilde{n}$ sentences. During the keyfact alignment process, each keyfact may be aligned with one or more summary sentences, or may remain unaligned. Let $\hat{n}_*$ denote the number of keyfacts that can be aligned with at least one summary sentence, and $\tilde{n}_*$ denote the number of summary sentences that can be aligned with at least one keyfact. The completeness and conciseness scores are defined as $S_{\text{com}}(\tilde{Y}) = S(\tilde{Y}|X, D = \text{com}) = \hat{n}_*/\hat{n}$ and $S_{\text{con}}(\tilde{Y}) = S(\tilde{Y}|X, D = \text{con}) = \tilde{n}_*/\tilde{n}$, respectively. These two fractions are intuitive proxy for summary quality, because completeness is measured as the proportion of document keyfacts that recoverable from the summary, indicating how much of the essential source content is preserved. Conversely, conciseness is measured as the proportion of summary sentences that are aligned with keyfacts, capturing how much of the summary is semantically meaningful rather than redundant or off-topic. Together, they reflect a balance between including sufficient information and avoiding unnecessary verbosity. Although completeness and conciseness are central to the utility of a summary, faithfulness, *i.e.*, whether the summary is factually consistent with the document constitutes a complementary but distinct dimension and is not directly assessed by these alignment-based scores.

C Prompt for FineSurE Calculation

Prompt for keyfact extraction:

You will be provided with a summary. Your task is to decompose the summary into a set of "key facts". A "key fact" is a single fact written as briefly and clearly as possible, encompassing at most 2-3 entities.
Here are nine examples of key facts to illustrate the desired level of granularity:
∗ Kevin Carr set off on his journey from Haytor.
∗ Kevin Carr set off on his journey from Dartmoor.
∗ Kevin Carr set off on his journey in July 2013.
∗ Kevin Carr is less than 24 hours away from completing his trip.
∗ Kevin Carr ran around the world unsupported.
∗ Kevin Carr ran with his tent.
∗ Kevin Carr is set to break the previous record.
∗ Kevin Carr is set to break the record by 24 hours.
∗ The previous record was held by an Australian.
Instruction:
First, read the summary carefully.
Second, decompose the summary into (at most 16) key facts.
Provide your answer in JSON format. The answer should be a dictionary with the key "key facts" containing the key facts as a list:
{ "key facts": ["first key fact", "second key fact", "third key fact"], Summary: [summary] }

Prompt for keyfact alignment:

You will receive a summary and a set of key facts for the same transcript. Your task is to assess if each key fact is inferred from the summary.
Instruction: First, compare each key fact with the summary. Second, check if the key fact is inferred from the summary and then response "Yes" or "No" for each key fact. If "Yes", specify the line number(s) of the summary sentence(s) relevant to each key fact.
Provide your answer in JSON format. The answer should be a list of dictionaries whose keys are "key fact", "response", and "line number": ["key fact": "first key fact", "response": "Yes", "line number": [1], "key fact": "second key fact", "response": "No", "line number": [], "key fact": "third key fact", "response": "Yes", "line number": [1, 2, 3]]
Summary: [summary] [N] key facts: [key-facts]

Text in blue indicates the part of the input that is fed into the prompt.

D Prompt for Control Generation

Prompt to control model generation prioritizing completeness over conciseness:

Below is an instruction that describes a task. Write a response that appropriately completes the request. Instruction: Please summarize the input document, prioritizing completeness over conciseness. Return the summary in the following JSON format: "Summary": "answer" Input:[document] Response:

Prompt to control model generation prioritizing conciseness over completeness:

Below is an instruction that describes a task. Write a response that appropriately completes the request. Instruction: Please summarize the input document, prioritizing conciseness over completeness. Return the summary in the following JSON format: "Summary": "answer" Input:[document] Response:

Prompt to control model generation balance completeness and conciseness:

Below is an instruction that describes a task. Write a response that appropriately completes the request. Instruction: Please summarize the input document, balancing completeness with conciseness. Return the summary in the following JSON format: "Summary": "answer" Input:[document] Response:

Sometimes, the model may fail to provide the correct JSON format, making it difficult to extract the intended answer. In such cases, it is often necessary to query the model multiple times to obtain

a valid JSON output. Text in blue indicates the part of the input that is fed into the prompt.

E Model Hyperparameters

For training, we selected $K = 15$, and used a batch size of 4, trained for 3 epochs. We set the hyperparameters as $\lambda = 0.5$, $\phi = 0.1$, $\gamma = 1$, and $\beta = 1$. For LoRA Training, we used lora_alpha = 16, lora_dropout = 0, and target_modules= {["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj"]}. For the model generation, we used nucleus (top-p) sampling "generation_config": GenerationConfig "do_sample": true, "temperature": 0.6, "top_p": 0.9.

F Trade-off verification

To verify that completeness and conciseness exhibit a clear trade-off relationship. We used the FeedSum dataset, which contains 13,348 document–summary pairs along with their completeness and conciseness scores. These scores were computed by G-Eval+ and FineSurE. We observe that 67% of the data points fall outside the extreme cases (1,1) and (0,0), suggesting meaningful variation between the two dimensions.

For these non-extreme points, we wanted to examine how one score behaves when the other becomes high. We filtered samples where either completeness or conciseness is above a certain threshold, and then computed the Spearman correlation between the two metrics.

From Table 5, we can clearly see that as one dimension approaches a higher score, the Spearman correlation becomes more negative, indicating a stronger trade-off. For G-Eval+ setting, the spearman correlation drops from –0.19 → –0.95. For FineSurE, it drops from –0.04 → –0.85.

In addition to the Spearman correlation analysis presented above, we further performed a Pareto frontier analysis to make this relationship explicit.

Based on this analysis, we obtained the following results: using Completeness and Conciseness scores from G-Eval+, the Pareto-optimal region lies in Completeness ∈ [0.800, 1.000] and Conciseness ∈ [0.800, 1.000], with an average trade-off slope of –1.000, meaning that for every 1-unit increase in Completeness, Conciseness decreases by approximately 1.000 units. Using FineSurE scores, the Pareto-optimal region lies in Completeness ∈ [0.938, 1.000] and Conciseness ∈ [0.889, 1.000], with an average trade-off slope of –1.778, meaning that for every 1-unit increase in Completeness, Conciseness decreases by roughly 1.778 units. Both analyses clearly reveal a negative trade-off. Because this trade-off is substantial and well-defined, the problem addressed in our paper is substantiated and deemed of research value.

Threshold	Valid Docs	Spearman Mean	Spearman Std
G-Eval+			
0.5	10116	-0.19±0.50	0.50
0.6	10116	-0.19±0.50	0.50
0.7	9427	-0.41±0.40	0.40
0.8	9427	-0.41±0.40	0.40
0.9	7809	-0.95±0.08	0.08
FineSurE			
0.5	8515	-0.04±0.47	0.47
0.6	7335	-0.27±0.51	0.51
0.7	5633	-0.48±0.51	0.51
0.8	4196	-0.60±0.47	0.47
0.9	1916	-0.85±0.13	0.13

Table 5: Spearman correlation between completeness and conciseness at different thresholds for non-extreme points in the FeedSum dataset (13,348 document-summary pairs). The mean and standard deviation are reported for each threshold.

Model	Pear Med	Pear IQR	Ked Med	Ked IQR
LLaMA	0.034	(-0.244, 0.291)	0.013	(-0.244, 0.166)
SummLLaMA	-0.037	(-0.346, 0.304)	0.035	(-0.182, 0.297)
LLaMA*	0.195	(-0.146, 0.459)	0.114	(-0.164, 0.295)
Qwen	0.027	(-0.299, 0.305)	-0.034	(-0.262, 0.210)
Qwen*	0.236	(-0.021, 0.488)	0.175	(-0.024, 0.349)
Mistral	0.003	(-0.240, 0.221)	-0.018	(-0.177, 0.138)
Mistral*	0.109	(-0.211, 0.396)	0.066	(-0.148, 0.304)

Table 6: Extended results including both Pearson and Kendall correlation metrics. Pear and Ked indicates Pearson and Kendall correlation metrics, respectively. Med indicates Medians. * indicates our fine-tuned version. Fine-tuned models consistently outperform their respective baselines across all three model families (LLaMA, Qwen, and Mistral). For example, LLaMA* achieves a Pearson correlation of 0.195 versus 0.034 for LLaMA and –0.037 for SummLLaMA; for Kendall-tau, LLaMA* obtains 0.114 versus 0.013 for LLaMA and 0.035 for SummLLaMA. The best performance is achieved by our fine-tuned Qwen*. These additional findings further strengthen our conclusions.

Dataset	Domain	Training	Test
FeedSum	lifestyle news daily life conversation	12000	600
OpoSum	product reviews	-	200
MeQSum	medical	-	200

Table 7: Dataset summary.

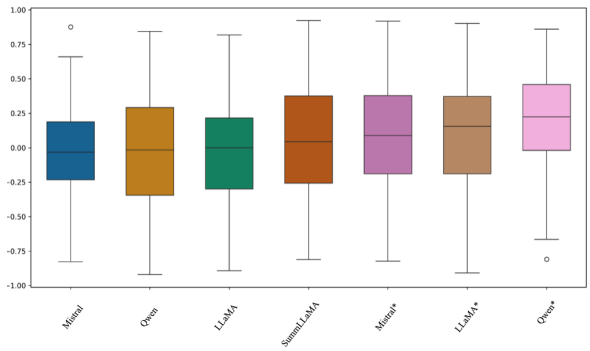

Figure 5: Distribution of Spearman correlations between model likelihood and model-based scores across different models. Models are sorted from left to right by median correlation value, from lowest to highest.

Model	**Criteria**	$\mathbf{S_{com}}$	$\mathbf{S_{con}}$	$\mathbf{HM}(\tilde{Y})$	$\mathbf{R}(\tilde{Y})$	**Succ rate**	**Prop. of 1**
LLaMA	$\text{Com}_\uparrow$	0.73 ± 0.24	0.52 ± 0.24	0.57 ± 0.18	0.38 ± 0.56	0.77	0.08
LLaMA	$\text{Con}_\uparrow$	0.57 ± 0.23	0.76 ± 0.26	0.60 ± 0.17	0.31 ± 0.60	0.65	0.22
LLaMA	Bal	0.70 ± 0.26	0.69 ± 0.27	0.66 ± 0.22	0.03 ± 0.51	0.84	-
SummLLaMA	$\text{Com}_\uparrow$	0.63 ± 0.21	0.73 ± 0.28	0.62 ± 0.17	-0.11 ± 0.66	0.65	0.24
SummLLaMA	$\text{Con}_\uparrow$	0.46 ± 0.18	0.99 ± 0.07	0.60 ± 0.17	0.86 ± 0.47	0.62	0.26
SummLLaMA	Bal	0.68 ± 0.27	0.97 ± 0.12	0.76 ± 0.21	-0.44 ± 0.49	0.89	-
LLaMA*(MS)	$\text{Com}_\uparrow$	0.75 ± 0.23	0.54 ± 0.21	0.60 ± 0.17	0.35 ± 0.51	0.78	0.07
LLaMA*(MS)	$\text{Con}_\uparrow$	0.56 ± 0.23	0.76 ± 0.26	0.60 ± 0.17	0.33 ± 0.60	0.66	0.23
LLaMA*(MS)	Bal	0.71 ± 0.25	0.70 ± 0.26	0.67 ± 0.22	0.02 ± 0.50	0.86	-
LLaMA*(CO)	$\text{Com}_\uparrow$	0.74 ± 0.23	0.53 ± 0.23	0.58 ± 0.17	0.36 ± 0.56	0.78	0.10
LLaMA*(CO)	$\text{Con}_\uparrow$	0.56 ± 0.26	0.77 ± 0.26	0.60 ± 0.18	0.33 ± 0.59	0.65	0.24
LLaMA*(CO)	Bal	0.72 ± 0.25	0.70 ± 0.27	0.67 ± 0.22	0.04 ± 0.52	0.88	-
LLaMA*(MR+MS)	$\text{Com}_\uparrow$	0.71 ± 0.22	0.63 ± 0.25	0.62 ± 0.16	0.15 ± 0.57	0.70	0.19
LLaMA*(MR+MS)	$\text{Con}_\uparrow$	0.53 ± 0.20	0.84 ± 0.24	0.63 ± 0.17	0.57 ± 0.52	0.57	0.31
LLaMA*(MR+MS)	Bal	0.73 ± 0.25	0.84 ± 0.24	0.75 ± 0.22	-0.14 ± 0.47	0.89	-
LLaMA*(MR+CO)	$\text{Com}_\uparrow$	0.74 ± 0.21	0.60 ± 0.24	0.61 ± 0.16	0.25 ± 0.53	0.71	0.18
LLaMA*(MR+CO)	$\text{Con}_\uparrow$	0.53 ± 0.19	0.88 ± 0.22	0.62 ± 0.16	0.54 ± 0.51	0.59	0.29
LLaMA*(MR+CO)	Bal	0.74 ± 0.25	0.80 ± 0.25	0.73 ± 0.22	-0.09 ± 0.47	0.90	-
LLaMA*(MS+CO)	$\text{Com}_\uparrow$	0.75 ± 0.23	0.53 ± 0.22	0.58 ± 0.18	0.38 ± 0.52	0.80	0.10
LLaMA*(MS+CO)	$\text{Con}_\uparrow$	0.55 ± 0.22	0.77 ± 0.26	0.60 ± 0.18	0.36 ± 0.59	0.63	0.24
LLaMA*(MS+CO)	Bal	0.72 ± 0.25	0.72 ± 0.26	0.69 ± 0.21	0.01 ± 0.51	0.88	-
LLaMA*(MR+MS+CO)	$\text{Com}_\uparrow$	0.72 ± 0.22	0.64 ± 0.25	0.63 ± 0.17	0.15 ± 0.55	0.71	0.19
LLaMA*(MR+MS+CO)	$\text{Con}_\uparrow$	0.52 ± 0.19	0.92 ± 0.18	0.63 ± 0.16	0.62 ± 0.53	0.60	0.31
LLaMA*(MR+MS+CO)	Bal	0.75 ± 0.25	0.85 ± 0.23	0.76 ± 0.21	-0.15 ± 0.48	0.92	-

Table 8: Ablation study on different loss components in LLaMA fine-tuning. We compare model performance across variants of LLaMA fine-tuned with different combinations of loss components using FineSurE. The reported values are averages across all FeedSum cases, with standard deviations in parentheses. $*$ indicates the base LLaMA model fine-tuned with our full objective. $\text{Com}_\uparrow$ prioritizes completeness, $\text{Con}_\uparrow$ prioritizes conciseness, while Bal aims to strike a balance between the two objectives. Succ rate denotes the percentage of examples where either S_{com} or S_{con} is non-zero. Proportion of 1 refers to cases where both S_{com} and S_{con} are exactly 1. These are considered uncontrollable in the Com$\uparrow$ and Con$\uparrow$ settings and are excluded when calculating S_{com}, S_{con}, $\text{HM}(\tilde{Y})$, and $\text{R}(\tilde{Y})$. Under the Bal objective, however, such cases are regarded as ideal and thus retained in evaluation (denoted as "–"). The MR-only variant is excluded here due to frequent repetition issues it introduces during generation.

Model	Category	Consistency	Coherence	Fluency	Relevance	ROUGE-1	ROUGE-2	ROUGE-3	BERTScore
LLaMA	Com$_\uparrow$	3.50 ± 1.19	2.11 ± 1.31	0.84 ± 0.78	3.06 ± 1.25	0.23 ± 0.13	0.08 ± 0.08	0.04 ± 0.06	0.30 ± 0.15
	Con$_\uparrow$	2.87 ± 0.88	1.58 ± 1.28	0.44 ± 0.55	2.49 ± 1.25	0.28 ± 0.14	0.09 ± 0.09	0.04 ± 0.06	0.29 ± 0.17
	Bal	3.29 ± 1.30	1.91 ± 1.32	0.63 ± 0.73	2.76 ± 1.25	0.27 ± 0.14	0.09 ± 0.09	0.04 ± 0.06	0.30 ± 0.15
SummLLaMA	Com$_\uparrow$	3.53 ± 1.21	2.02 ± 1.24	0.72 ± 0.74	3.00 ± 1.26	0.23 ± 0.13	0.08 ± 0.07	0.04 ± 0.05	0.31 ± 0.13
	Con$_\uparrow$	2.48 ± 0.79	1.30 ± 1.08	0.33 ± 0.44	2.22 ± 1.25	0.28 ± 0.14	0.09 ± 0.10	0.04 ± 0.07	0.27 ± 0.17
	Bal	3.34 ± 1.15	1.67 ± 1.20	0.45 ± 0.55	2.58 ± 1.22	0.27 ± 0.14	0.10 ± 0.09	0.04 ± 0.06	0.29 ± 0.15
LLaMA*	Com$_\uparrow$	3.48 ± 1.24	2.08 ± 1.28	0.85 ± 0.78	3.19 ± 1.27	0.23 ± 0.12	0.08 ± 0.07	0.04 ± 0.05	0.31 ± 0.14
	Con$_\uparrow$	2.89 ± 0.83	1.63 ± 1.21	0.49 ± 0.53	2.61 ± 1.27	0.28 ± 0.14	0.09 ± 0.09	0.04 ± 0.06	0.28 ± 0.16
	Bal	3.38 ± 1.17	1.88 ± 1.22	0.65 ± 0.72	2.92 ± 1.25	0.27 ± 0.13	0.10 ± 0.08	0.04 ± 0.06	0.30 ± 0.15

Table 9: Detailed comparison of model performance across multiple quality dimensions. We evaluate consistency, coherence, fluency, and relevance using G-Eval, reporting averages over all cases with standard deviations. ROUGE and BERTScore are also included, computed based on human reference summaries. $*$ denotes foundation models fine-tuned using our method. These metrics provide a more comprehensive assessment of summary quality beyond traditional reference-based measures.

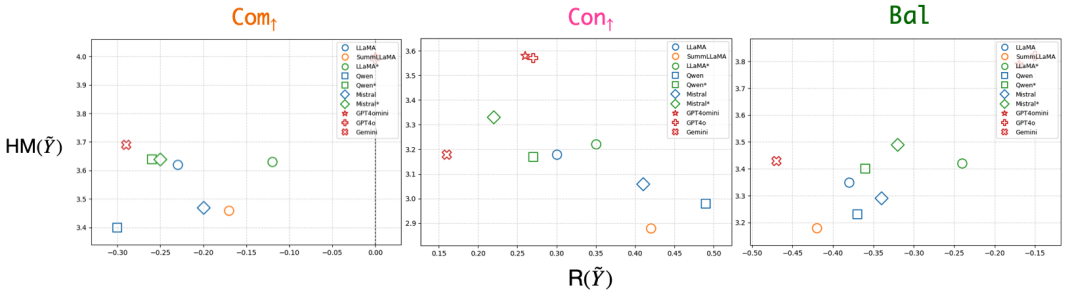

Figure 6: Scatter plot of model performance across different control settings using G-Eval+. Each point represents the mean of HM($\tilde{Y}$) (y-axis) and R($\tilde{Y}$) (x-axis) across all test cases in FeedSum test set. Models are grouped by color into four categories: Baseline models (blue), Our methods (green), SummLLaMA (orange), and Commercial models (red). The three panels show performance under different control priorities: Com$_\uparrow$ prioritizes completeness, Con$_\uparrow$ prioritizes conciseness, and Bal aims to balance both. The vertical dashed line at R($\tilde{Y}$) = 0 represents the target equilibrium between completeness and conciseness, providing a reference for model controllability.

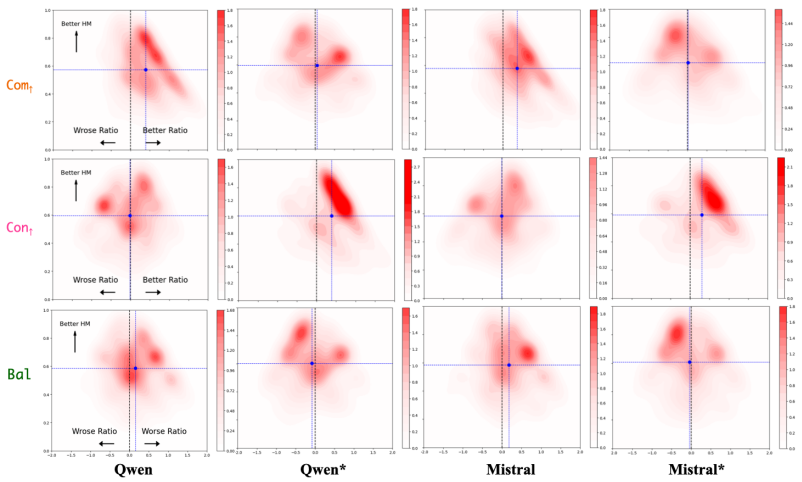

Figure 7: Contour plot showing the distribution of models with respect to control ability (x-axis: R($\tilde{Y}$)) and summary quality (y-axis: HM($\tilde{Y}$)). Density is represented by color, highlighting regions of strong performance.

Model	Criteria	$\mathbf{S_{com}}$	$\mathbf{S_{con}}$	$\mathbf{HM}(\tilde{Y})$	$\mathbf{R}(\tilde{Y})$	Succ rate	Prop. of 1
	Com$_\uparrow$	0.73 ± 0.24	0.52 ± 0.24	0.57 ± 0.18	0.38 ± 0.56	0.77	0.08
LLaMA	Con$_\uparrow$	0.57 ± 0.23	0.76 ± 0.26	0.60 ± 0.17	0.31 ± 0.6	0.65	0.22
	Bal	0.70 ± 0.26	0.69 ± 0.27	0.66 ± 0.22	0.03 ± 0.51	0.84	-
	Com$_\uparrow$	0.63 ± 0.21	0.73 ± 0.28	0.62 ± 0.17	-0.11 ± 0.66	0.65	0.24
SummLLaMA	Con$_\uparrow$	0.46 ± 0.18	0.99 ± 0.07	0.60 ± 0.17	0.86 ± 0.47	0.62	0.26
	Bal	0.68 ± 0.27	0.97 ± 0.12	0.76 ± 0.21	-0.44 ± 0.49	0.89	-
	Com$_\uparrow$	0.72 ± 0.22	0.64 ± 0.25	0.63 ± 0.17	0.15 ± 0.55	0.71	0.19
LLaMA*	Con$_\uparrow$	0.52 ± 0.19	0.92 ± 0.18	0.63 ± 0.16	0.62 ± 0.53	0.60	0.31
	Bal	0.75 ± 0.25	0.85 ± 0.23	0.76 ± 0.21	-0.15 ± 0.48	0.92	-
	Com$_\uparrow$	0.74 ± 0.23	0.52 ± 0.21	0.57 ± 0.17	0.40 ± 0.52	0.81	0.08
Qwen	Con$_\uparrow$	0.64 ± 0.24	0.64 ± 0.24	0.60 ± 0.17	-0.01 ± 0.53	0.71	0.16
	Bal	0.72 ± 0.25	0.64 ± 0.25	0.65 ± 0.21	0.13 ± 0.49	0.88	-
	Com$_\uparrow$	0.66 ± 0.23	0.63 ± 0.25	0.60 ± 0.16	0.05 ± 0.6	0.70	0.19
Qwen*	Con$_\uparrow$	0.54 ± 0.21	0.78 ± 0.25	0.60 ± 0.18	0.39 ± 0.53	0.63	0.24
	Bal	0.71 ± 0.26	0.75 ± 0.26	0.70 ± 0.22	-0.06 ± 0.48	0.87	-
	Com$_\uparrow$	0.73 ± 0.23	0.53 ± 0.22	0.58 ± 0.18	0.36 ± 0.53	0.79	0.08
Mistral	Con$_\uparrow$	0.63 ± 0.24	0.63 ± 0.26	0.59 ± 0.17	-0.01 ± 0.57	0.70	0.18
	Bal	0.73 ± 0.25	0.64 ± 0.26	0.64 ± 0.22	0.15 ± 0.51	0.88	-
	Com$_\uparrow$	0.67 ± 0.24	0.68 ± 0.27	0.62 ± 0.18	0.02 ± 0.64	0.63	0.20
Mistral*	Con$_\uparrow$	0.56 ± 0.22	0.75 ± 0.27	0.59 ± 0.17	0.30 ± 0.6	0.63	0.24
	Bal	0.71 ± 0.26	0.74 ± 0.27	0.69 ± 0.23	-0.03 ± 0.5	0.87	-
	Com$_\uparrow$	0.83 ± 0.21	0.48 ± 0.2	0.58 ± 0.18	0.6 ± 0.51	0.84	0.04
GPT4omini	Con$_\uparrow$	0.67 ± 0.23	0.64 ± 0.25	0.61 ± 0.17	-0.06 ± 0.53	0.69	0.2
	Bal	0.78 ± 0.23	0.62 ± 0.25	0.66 ± 0.21	0.29 ± 0.51	0.89	-
	Com$_\uparrow$	0.82 ± 0.21	0.47 ± 0.2	0.57 ± 0.18	0.62 ± 0.49	0.85	0.06
GPT4o	Con$_\uparrow$	0.66 ± 0.23	0.61 ± 0.24	0.59 ± 0.17	-0.08 ± 0.53	0.69	0.19
	Bal	0.78 ± 0.23	0.62 ± 0.26	0.66 ± 0.22	0.27 ± 0.52	0.89	-
	Com$_\uparrow$	0.72 ± 0.24	0.56 ± 0.22	0.59 ± 0.18	0.27 ± 0.51	0.78	0.12
Keyfact_GPT4o	Con$_\uparrow$	0.56 ± 0.23	0.71 ± 0.26	0.58 ± 0.18	0.26 ± 0.56	0.62	0.24
	Bal	0.70 ± 0.27	0.74 ± 0.26	0.69 ± 0.23	-0.06 ± 0.47	0.87	-
	Com$_\uparrow$	0.73 ± 0.21	0.63 ± 0.27	0.63 ± 0.17	0.14 ± 0.50	0.71	0.20
Keyfact_GPT4o*	Con$_\uparrow$	0.52 ± 0.26	0.93 ± 0.26	0.63 ± 0.18	0.65 ± 0.50	0.60	0.32
	Bal	0.74 ± 0.23	0.84 ± 0.21	0.75 ± 0.25	-0.16 ± 0.51	0.91	-
	Com$_\uparrow$	0.88 ± 0.17	0.61 ± 0.2	0.69 ± 0.15	0.42 ± 0.45	0.77	0.18
GPT4-turbo	Con$_\uparrow$	0.69 ± 0.2	0.82 ± 0.21	0.71 ± 0.14	0.18 ± 0.49	0.54	0.42
	Bal	0.86 ± 0.19	0.81 ± 0.22	0.81 ± 0.18	0.07 ± 0.38	0.95	-
	Com$_\uparrow$	0.86 ± 0.2	0.45 ± 0.19	0.57 ± 0.18	0.71 ± 0.49	0.86	0.03
Gemini	Con$_\uparrow$	0.61 ± 0.24	0.66 ± 0.25	0.59 ± 0.16	0.08 ± 0.58	0.67	0.22
	Bal	0.76 ± 0.24	0.66 ± 0.26	0.67 ± 0.21	0.16 ± 0.49	0.88	-

Table 10: Detailed comparison of model performance using FineSurE. The value shows averages over all cases in FeedSum with standard deviations. $*$ indicates the foundation model fine-tuned using our method. Com$_\uparrow$ prioritizes completeness, Con$_\uparrow$ prioritizes conciseness, and Bal aims to balance both. *Succ rate* denotes the proportion of cases where either S_{com} or S_{con} is non-zero. *Proportion of 1* indicates the proportion of cases where both S_{com} and S_{con} equal 1. In the Com$_\uparrow$ and Con$_\uparrow$ settings, these are considered uncontrollable cases and are excluded when computing S_{com}, S_{con}, HM$(\tilde{Y})$, and R$(\tilde{Y})$. In contrast, under Bal, these cases are treated as ideal (i.e., perfect), so they are retained (denoted as "–"). Key_GPT4o is a prompt-oriented baseline. In Key_GPT4o, we first use GPT-4o to extract key facts, which are then incorporated into a subsequent prompt for the summarizer. Key_GPT4o* is a single-pass variant, where we prompt the model to first generate key facts and then produce a summary conditioned on these key facts within a single prompt. It is worth noting that if the model used for model-based scoring shares the same architecture as the model being evaluated (*e.g.*, using FineSurE(GPT-4o) to evaluate GPT-4o), the resulting score may be biased. In such cases, model-based scoring tends to favor models with similar architectures (Wataoka et al., 2024; Xu et al., 2024).

Model	Criteria	Faithfulness	AvgFaith
LLaMA	Com$_\uparrow$	0.83 ± 0.26	0.82
	Con$_\uparrow$	0.83 ± 0.33	
	Bal	0.8 ± 0.32	
SummLLaMA	Com$_\uparrow$	0.85 ± 0.32	0.85
	Con$_\uparrow$	0.85 ± 0.35	
	Bal	0.84 ± 0.36	
LLaMA*	Com$_\uparrow$	0.85 ± 0.3	0.82
	Con$_\uparrow$	0.8 ± 0.42	
	Bal	0.8 ± 0.38	
Qwen	Com$_\uparrow$	0.88 ± 0.2	0.85
	Con$_\uparrow$	0.83 ± 0.29	
	Bal	0.85 ± 0.25	
Qwen*	Com$_\uparrow$	0.87 ± 0.31	0.85
	Con$_\uparrow$	0.83 ± 0.37	
	Bal	0.85 ± 0.32	
Mistral	Com$_\uparrow$	0.91 ± 0.17	0.9
	Con$_\uparrow$	0.89 ± 0.24	
	Bal	0.9 ± 0.22	
Mistral*	Com$_\uparrow$	0.81 ± 0.4	0.81
	Con$_\uparrow$	0.82 ± 0.41	
	Bal	0.8 ± 0.41	
GPT4omini	Com$_\uparrow$	0.95 ± 0.13	0.94
	Con$_\uparrow$	0.93 ± 0.19	
	Bal	0.93 ± 0.16	
GPT4o	Com$_\uparrow$	0.94 ± 0.14	0.93
	Con$_\uparrow$	0.91 ± 0.22	
	Bal	0.95 ± 0.14	
GPT4-turbo	Com$_\uparrow$	0.95 ± 0.12	0.93
	Con$_\uparrow$	0.92 ± 0.19	
	Bal	0.92 ± 0.2	
Gemini	Com$_\uparrow$	0.96 ± 0.12	0.93
	Con$_\uparrow$	0.89 ± 0.24	
	Bal	0.93 ± 0.18	

Table 11: Faithfulness evaluation results using FineSurE across all models. The value shows averages over all cases in FeedSum with standard deviations. AvgFaith indicates the average of faithfulness across different criteria.

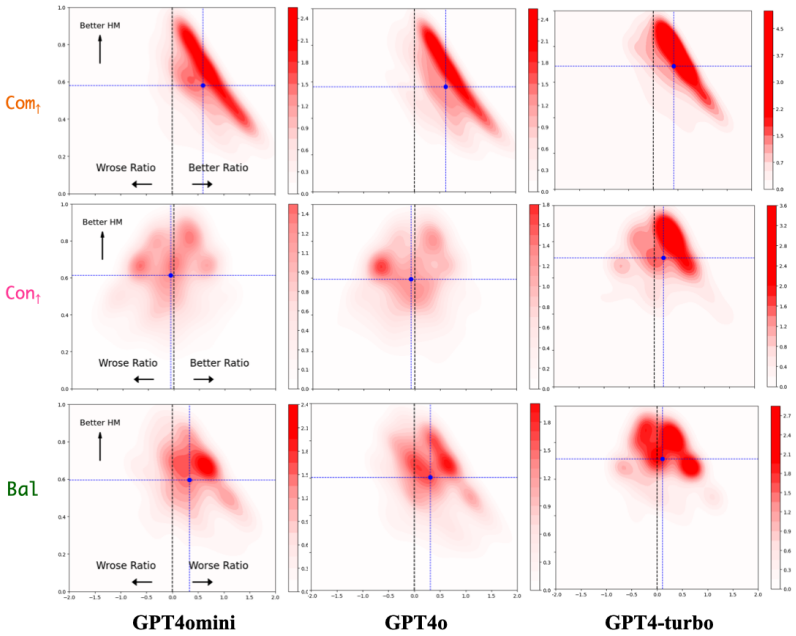

Figure 8: Contour plot showing the distribution of models with respect to control ability (x-axis: R($\tilde{Y}$)) and summary quality (y-axis: HM($\tilde{Y}$)). Density is represented by color, highlighting regions of strong performance.

Model	Criteria	S_{com}	S_{con}	HM($\tilde{Y}$)	R($\tilde{Y}$)	Faithfulness	AvgFaith
	Com$_\uparrow$	3.51 ± 0.93	3.92 ± 0.59	3.62 ± 0.71	-0.23 ± 0.37	0.88 ± 0.17	
LLaMA	Con$_\uparrow$	2.79 ± 0.9	3.95 ± 0.67	3.18 ± 0.76	0.3 ± 0.43	0.91 ± 0.2	0.89
	Bal	3.06 ± 0.92	3.94 ± 0.65	3.35 ± 0.76	-0.38 ± 0.39	0.88 ± 0.21	
	Com$_\uparrow$	3.43 ± 0.89	3.66 ± 0.64	3.46 ± 0.69	-0.17 ± 0.36	0.94 ± 0.14	
SummLLaMA	Con$_\uparrow$	2.4 ± 0.82	3.86 ± 0.71	2.88 ± 0.73	0.42 ± 0.4	0.94 ± 0.15	0.94
	Bal	2.84 ± 0.84	3.82 ± 0.65	3.18 ± 0.71	-0.42 ± 0.37	0.95 ± 0.13	
	Com$_\uparrow$	3.54 ± 0.89	3.87 ± 0.59	3.63 ± 0.67	-0.12 ± 0.32	0.91 ± 0.21	
LLaMA*	Con$_\uparrow$	2.86 ± 0.86	3.9 ± 0.67	3.22 ± 0.71	0.35 ± 0.4	0.91 ± 0.27	0.91
	Bal	3.19 ± 0.87	3.89 ± 0.6	3.42 ± 0.69	-0.24 ± 0.36	0.9 ± 0.25	
	Com$_\uparrow$	3.1 ± 0.9	4 ± 0.62	3.4 ± 0.73	-0.3 ± 0.39	0.9 ± 0.22	
Qwen	Con$_\uparrow$	2.5 ± 0.77	3.93 ± 0.73	2.98 ± 0.69	0.49 ± 0.38	0.91 ± 0.24	0.9
	Bal	2.87 ± 0.89	3.95 ± 0.7	3.23 ± 0.76	-0.37 ± 0.43	0.9 ± 0.24	
	Com$_\uparrow$	3.48 ± 0.9	3.99 ± 0.56	3.64 ± 0.7	-0.26 ± 0.41	0.91 ± 0.14	
Qwen*	Con$_\uparrow$	2.82 ± 0.87	3.89 ± 0.73	3.17 ± 0.74	0.27 ± 0.41	0.91 ± 0.18	0.91
	Bal	3.13 ± 0.94	3.95 ± 0.69	3.4 ± 0.77	-0.36 ± 0.41	0.9 ± 0.16	
	Com$_\uparrow$	3.47 ± 0.85	4 ± 0.55	3.64 ± 0.63	-0.25 ± 0.32	0.92 ± 0.23	
Mistral	Con$_\uparrow$	3.02 ± 0.89	3.94 ± 0.68	3.33 ± 0.76	0.22 ± 0.42	0.92 ± 0.25	0.92
	Bal	3.24 ± 0.88	3.98 ± 0.63	3.49 ± 0.7	-0.32 ± 0.36	0.92 ± 0.25	
	Com$_\uparrow$	3.26 ± 0.88	3.89 ± 0.66	3.47 ± 0.69	-0.2 ± 0.32	0.91 ± 0.14	
Mistral*	Con$_\uparrow$	2.66 ± 0.89	3.83 ± 0.78	3.06 ± 0.79	0.41 ± 0.36	0.92 ± 0.17	0.92
	Bal	2.93 ± 0.89	3.96 ± 0.66	3.29 ± 0.73	-0.34 ± 0.36	0.93 ± 0.14	
	Com$_\uparrow$	4.08 ± 0.77	4.03 ± 0.47	4 ± 0.54	0 ± 0.26	0.97 ± 0.09	
GPT4omini	Con$_\uparrow$	3.32 ± 0.97	4.13 ± 0.64	3.58 ± 0.77	0.26 ± 0.4	0.98 ± 0.08	0.98
	Bal	3.63 ± 0.92	4.16 ± 0.58	3.79 ± 0.71	-0.17 ± 0.37	0.99 ± 0.07	
	Com$_\uparrow$	4.09 ± 0.78	4 ± 0.44	3.99 ± 0.57	0 ± 0.29	0.97 ± 0.09	
GPT4o	Con$_\uparrow$	3.29 ± 0.91	4.14 ± 0.64	3.57 ± 0.74	0.27 ± 0.39	0.98 ± 0.1	0.98
	Bal	3.67 ± 0.88	4.15 ± 0.59	3.82 ± 0.69	-0.15 ± 0.36	0.98 ± 0.07	
	Com$_\uparrow$	3.91 ± 0.93	4.02 ± 0.48	3.88 ± 0.67	-0.06 ± 0.36	0.98 ± 0.08	
GPT4-turbo	Con$_\uparrow$	3.11 ± 0.99	4.05 ± 0.68	3.41 ± 0.85	0.33 ± 0.52	0.98 ± 0.09	0.98
	Bal	3.52 ± 0.96	4.08 ± 0.62	3.68 ± 0.76	-0.19 ± 0.42	0.98 ± 0.09	
	Com$_\uparrow$	3.7 ± 0.95	3.86 ± 0.52	3.69 ± 0.69	-0.29 ± 0.39	0.95 ± 0.09	
Gemini	Con$_\uparrow$	2.82 ± 0.95	3.94 ± 0.69	3.18 ± 0.82	0.16 ± 0.46	0.95 ± 0.11	0.95
	Bal	3.18 ± 0.92	3.96 ± 0.65	3.43 ± 0.74	-0.47 ± 0.41	0.95 ± 0.09	

Table 12: Detailed comparison of model performance using G-Eval+. The value shows averages over all cases in FeedSum with standard deviations. $*$ denotes foundation models fine-tuned using our method. Com$_\uparrow$ emphasizes completeness, Con$_\uparrow$ emphasizes conciseness, and Bal seeks to balance both. *Succ rate* refers to the proportion of cases where either S_{com} or S_{con} is non-zero. AvgFaith indicates the average of Faithfulness across different Criteria.

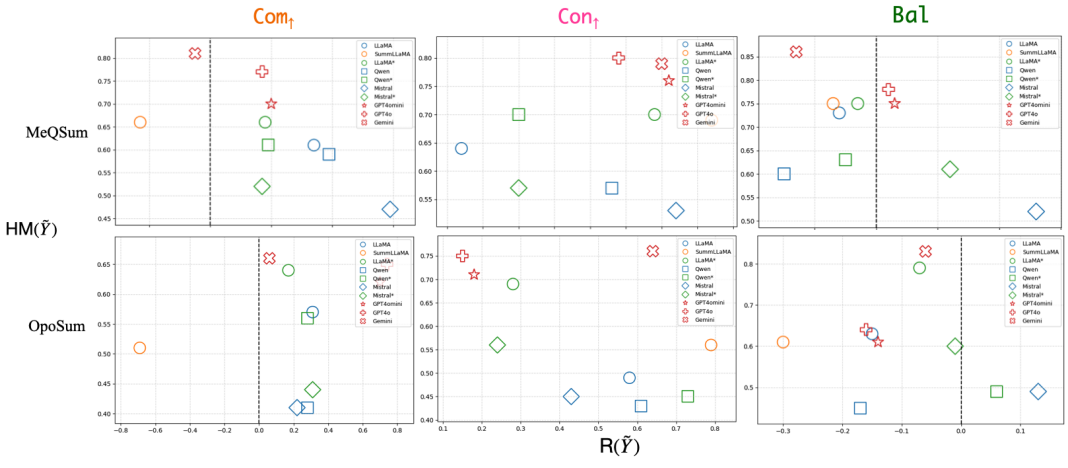

Figure 9: Scatter plot of model performance testing in out-of-domain data (MeQSum; OpoSum) across different control settings. Each point represents the mean of HM($\tilde{Y}$) (y-axis) and R($\tilde{Y}$) (x-axis) across all test cases in FeedSum test set. Models are grouped by color into four categories: Baseline models (blue), Our methods (green), SummLLaMA (orange), and Commercial models (red). The three panels show performance under different control priorities: Com$_\uparrow$ prioritizes completeness, Con$_\uparrow$ prioritizes conciseness, and Bal aims to balance both. The vertical dashed line at R($\tilde{Y}$) = 0 represents the target equilibrium between completeness and conciseness, providing a reference for model controllability.

Model	Source	Category	$\mathbf{S_{com}}$	$\mathbf{S_{con}}$	$\mathbf{HM}(\tilde{Y})$	$\mathbf{R}(\tilde{Y})$	Succ rate	Prop. of 1
LLaMA	meqsum	$\text{Com}_\uparrow$	0.74 ± 0.18	0.55 ± 0.23	0.61 ± 0.18	0.34 ± 0.54	0.72	0.24
		$\text{Con}_\uparrow$	0.66 ± 0.16	0.7 ± 0.16	0.64 ± 0.14	0.07 ± 0.39	0.78	0.18
		Bal	0.74 ± 0.2	0.8 ± 0.2	0.73 ± 0.18	-0.06 ± 0.39	0.98	-
	oposum	$\text{Com}_\uparrow$	0.69 ± 0.17	0.52 ± 0.18	0.57 ± 0.12	0.31 ± 0.37	0.61	0.32
		$\text{Con}_\uparrow$	0.4 ± 0.16	0.7 ± 0.17	0.49 ± 0.13	0.58 ± 0.41	0.86	0.11
		Bal	0.63 ± 0.19	0.75 ± 0.19	0.63 ± 0.16	-0.15 ± 0.33	0.94	-
SummLLaMA	meqsum	$\text{Com}_\uparrow$	0.61 ± 0.15	0.79 ± 0.06	0.66 ± 0.12	-0.23 ± 0.33	0.6	0.38
		$\text{Con}_\uparrow$	0.61 ± 0.17	0.9 ± 0.04	0.69 ± 0.14	0.42 ± 0.35	0.81	0.18
		Bal	0.75 ± 0.2	0.83 ± 0.02	0.75 ± 0.15	-0.07 ± 0.35	0.99	-
	oposum	$\text{Com}_\uparrow$	0.4 ± 0.19	0.83 ± 0	0.51 ± 0.18	-0.69 ± 0.44	0.11	0.01
		$\text{Con}_\uparrow$	0.43 ± 0.18	0.89 ± 0	0.56 ± 0.17	0.79 ± 0.43	0.16	0.02
		Bal	0.54 ± 0.26	0.75 ± 0	0.61 ± 0.22	-0.3 ± 0.51	0.1	-
LLaMA*	meqsum	$\text{Com}_\uparrow$	0.73 ± 0.16	0.63 ± 0.2	0.66 ± 0.16	0.18 ± 0.45	0.63	0.34
		$\text{Con}_\uparrow$	0.64 ± 0.17	0.85 ± 0.09	0.7 ± 0.16	0.34 ± 0.41	0.78	0.2
		Bal	0.79 ± 0.19	0.82 ± 0.12	0.75 ± 0.14	-0.03 ± 0.34	0.96	-
	oposum	$\text{Com}_\uparrow$	0.73 ± 0.16	0.63 ± 0.19	0.64 ± 0.12	0.17 ± 0.36	0.55	0.44
		$\text{Con}_\uparrow$	0.64 ± 0.14	0.83 ± 0.04	0.69 ± 0.11	0.28 ± 0.25	0.84	0.14
		Bal	0.78 ± 0.18	0.85 ± 0.1	0.79 ± 0.13	-0.07 ± 0.28	0.98	-
Qwen	meqsum	$\text{Com}_\uparrow$	0.76 ± 0.16	0.53 ± 0.19	0.59 ± 0.13	0.39 ± 0.44	0.58	0.4
		$\text{Con}_\uparrow$	0.53 ± 0.16	0.69 ± 0.14	0.57 ± 0.15	0.28 ± 0.45	0.68	0.3
		Bal	0.59 ± 0.19	0.7 ± 0.12	0.6 ± 0.15	-0.15 ± 0.35	0.98	-
	oposum	$\text{Com}_\uparrow$	0.5 ± 0.12	0.38 ± 0	0.41 ± 0.13	0.28 ± 0.35	0.1	0
		$\text{Con}_\uparrow$	0.35 ± 0.16	0.64 ± 0	0.43 ± 0.15	0.61 ± 0.38	0.13	0
		Bal	0.44 ± 0.26	0.53 ± 0	0.45 ± 0.22	-0.17 ± 0.52	0.09	-
Qwen*	meqsum	$\text{Com}_\uparrow$	0.7 ± 0.16	0.6 ± 0.09	0.61 ± 0.13	0.19 ± 0.36	0.62	0.36
		$\text{Con}_\uparrow$	0.73 ± 0.18	0.82 ± 0.04	0.7 ± 0.15	0.15 ± 0.39	0.8	0.17
		Bal	0.66 ± 0.2	0.71 ± 0.05	0.63 ± 0.15	-0.05 ± 0.35	0.98	-
	oposum	$\text{Com}_\uparrow$	0.69 ± 0.16	0.53 ± 0.18	0.56 ± 0.12	0.28 ± 0.35	0.74	0.26
		$\text{Con}_\uparrow$	0.35 ± 0.17	0.71 ± 0.22	0.45 ± 0.12	0.73 ± 0.45	0.64	0.34
		Bal	0.53 ± 0.16	0.51 ± 0.22	0.49 ± 0.16	0.06 ± 0.33	0.98	-
Mistral	meqsum	$\text{Com}_\uparrow$	0.67 ± 0.15	0.4 ± 0.19	0.47 ± 0.13	0.59 ± 0.43	0.57	0.42
		$\text{Con}_\uparrow$	0.5 ± 0.17	0.7 ± 0.15	0.53 ± 0.15	0.37 ± 0.43	0.69	0.3
		Bal	0.63 ± 0.19	0.5 ± 0.11	0.52 ± 0.15	0.26 ± 0.35	0.99	-
	oposum	$\text{Com}_\uparrow$	0.49 ± 0.16	0.39 ± 0.18	0.41 ± 0.12	0.22 ± 0.36	0.72	0.26
		$\text{Con}_\uparrow$	0.4 ± 0.16	0.6 ± 0.22	0.45 ± 0.12	0.43 ± 0.44	0.64	0.33
		Bal	0.55 ± 0.16	0.49 ± 0.22	0.49 ± 0.16	0.13 ± 0.32	0.98	-
Mistral*	meqsum	$\text{Com}_\uparrow$	0.63 ± 0.21	0.55 ± 0.04	0.52 ± 0.18	0.17 ± 0.49	0.65	0.28
		$\text{Con}_\uparrow$	0.56 ± 0.19	0.63 ± 0.03	0.57 ± 0.17	0.15 ± 0.47	0.79	0.18
		Bal	0.66 ± 0.21	0.6 ± 0.02	0.61 ± 0.16	0.12 ± 0.42	0.94	-
	oposum	$\text{Com}_\uparrow$	0.56 ± 0.14	0.42 ± 0.18	0.44 ± 0.11	0.31 ± 0.39	0.57	0.36
		$\text{Con}_\uparrow$	0.53 ± 0.15	0.66 ± 0.06	0.56 ± 0.12	0.24 ± 0.3	0.8	0.16
		Bal	0.61 ± 0.19	0.63 ± 0.07	0.6 ± 0.14	-0.01 ± 0.31	0.95	-
GPT4omini	meqsum	$\text{Com}_\uparrow$	0.79 ± 0.22	0.66 ± 0.25	0.7 ± 0.21	0.2 ± 0.57	0.4	0.46
		$\text{Con}_\uparrow$	0.65 ± 0.18	0.91 ± 0.18	0.76 ± 0.16	0.36 ± 0.43	0.61	0.21
		Bal	0.8 ± 0.24	0.78 ± 0.23	0.75 ± 0.22	0.03 ± 0.42	0.93	-
	oposum	$\text{Com}_\uparrow$	0.68 ± 0.18	0.62 ± 0.2	0.62 ± 0.13	0.7 ± 0.43	0.68	0.23
		$\text{Con}_\uparrow$	0.7 ± 0.19	0.79 ± 0.09	0.71 ± 0.15	0.18 ± 0.34	0.59	0.28
		Bal	0.59 ± 0.21	0.69 ± 0.1	0.61 ± 0.17	-0.14 ± 0.33	0.96	-
GPT4o	meqsum	$\text{Com}_\uparrow$	0.86 ± 0.18	0.73 ± 0.08	0.77 ± 0.15	0.17 ± 0.36	0.46	0.48
		$\text{Con}_\uparrow$	0.71 ± 0.19	0.94 ± 0.08	0.8 ± 0.17	0.29 ± 0.39	0.7	0.25
		Bal	0.83 ± 0.2	0.81 ± 0.18	0.78 ± 0.18	0.02 ± 0.47	0.97	-
	oposum	$\text{Com}_\uparrow$	0.71 ± 0.15	0.65 ± 0.19	0.65 ± 0.13	0.74 ± 0.31	0.71	0.25
		$\text{Con}_\uparrow$	0.75 ± 0.19	0.85 ± 0.23	0.75 ± 0.13	0.15 ± 0.45	0.62	0.32
		Bal	0.61 ± 0.16	0.73 ± 0.23	0.64 ± 0.17	-0.16 ± 0.33	0.98	-
GPT4-turbo	meqsum	$\text{Com}_\uparrow$	0.88 ± 0.14	0.79 ± 0.15	0.81 ± 0.12	0.13 ± 0.34	0.5	0.5
		$\text{Con}_\uparrow$	0.73 ± 0.15	0.97 ± 0.1	0.81 ± 0.12	0.31 ± 0.28	0.74	0.26
		Bal	0.84 ± 0.16	0.83 ± 0.08	0.79 ± 0.12	0.03 ± 0.25	1	-
	oposum	$\text{Com}_\uparrow$	0.73 ± 0.13	0.68 ± 0.15	0.67 ± 0.11	0.75 ± 0.27	0.73	0.27
		$\text{Con}_\uparrow$	0.77 ± 0.15	0.87 ± 0.21	0.76 ± 0.11	0.16 ± 0.43	0.64	0.34
		Bal	0.63 ± 0.14	0.75 ± 0.19	0.65 ± 0.14	-0.15 ± 0.29	0.99	-
Gemini	meqsum	$\text{Com}_\uparrow$	0.8 ± 0.24	0.86 ± 0.06	0.81 ± 0.17	-0.05 ± 0.37	0.63	0.46
		$\text{Con}_\uparrow$	0.69 ± 0.2	0.97 ± 0.24	0.79 ± 0.2	0.35 ± 0.49	0.78	0.22
		Bal	0.84 ± 0.2	0.98 ± 0.13	0.86 ± 0.2	-0.13 ± 0.45	0.66	-
	oposum	$\text{Com}_\uparrow$	0.71 ± 0.2	0.68 ± 0.21	0.66 ± 0.14	0.06 ± 0.4	0.76	0.14
		$\text{Con}_\uparrow$	0.77 ± 0.19	0.82 ± 0.25	0.76 ± 0.15	0.64 ± 0.48	0.6	0.28
		Bal	0.81 ± 0.18	0.87 ± 0.24	0.83 ± 0.2	-0.06 ± 0.36	0.64	-

Table 13: Detailed comparison of model performance using FineSurE. The value shows averages over all cases across two out-of-domain datasets with standard deviations. $*$ denotes foundation models fine-tuned using our method. $\text{Com}_\uparrow$ emphasizes completeness, $\text{Con}_\uparrow$ emphasizes conciseness, and Bal seeks to balance both. *Succ rate* refers to the proportion of cases where either S_{com} or S_{con} is non-zero. *Proportion of 1* indicates the proportion of cases where both S_{com} and S_{con} equal 1. In the $\text{Com}_\uparrow$ and $\text{Con}_\uparrow$ settings, such cases are considered uncontrollable and are excluded from the computation of S_{com}, S_{con}, $\text{HM}(\tilde{Y})$, and $\text{R}(\tilde{Y})$. In contrast, under the Bal setting, these cases are treated as ideal (i.e., perfect) and are retained (denoted as "–").

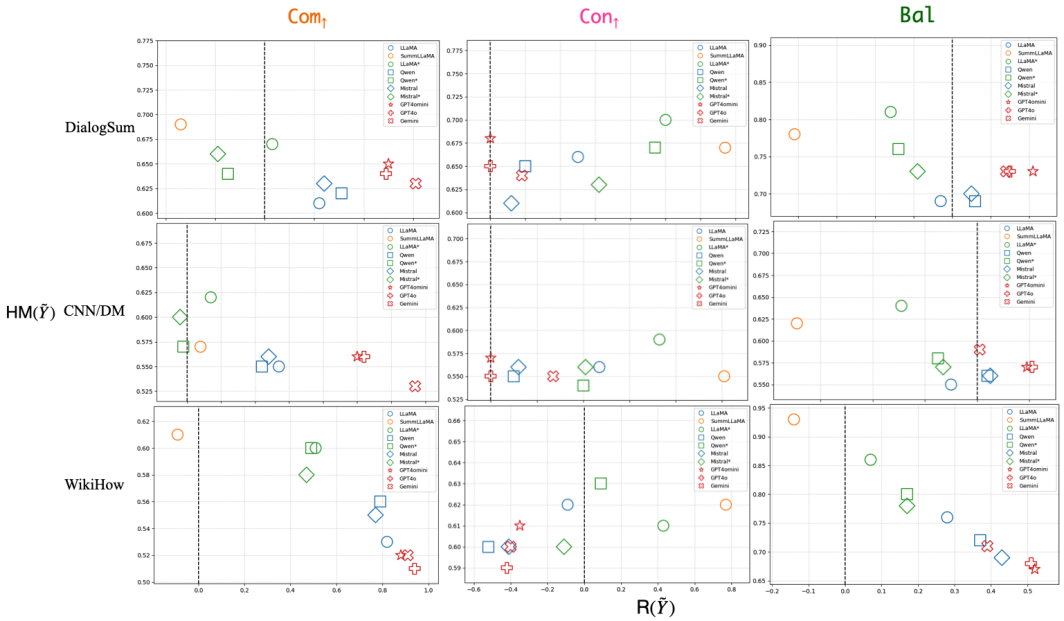

Figure 10: Scatter plot of model performance separated by different domain (DialogSum; CNN/DM; WikiHow) across different control settings. Each point represents the mean of HM($\tilde{Y}$) (y-axis) and R($\tilde{Y}$) (x-axis) across all test cases in FeedSum test set. Models are grouped by color into four categories: Baseline models (blue), Our methods (green), SummLLaMA (orange), and Commercial models (red). The three panels show performance under different control priorities: Com$_\uparrow$ prioritizes completeness, Con$_\uparrow$ prioritizes conciseness, and Bal aims to balance both. The vertical dashed line at R($\tilde{Y}$) = 0 represents the target equilibrium between completeness and conciseness, providing a reference for model controllability.

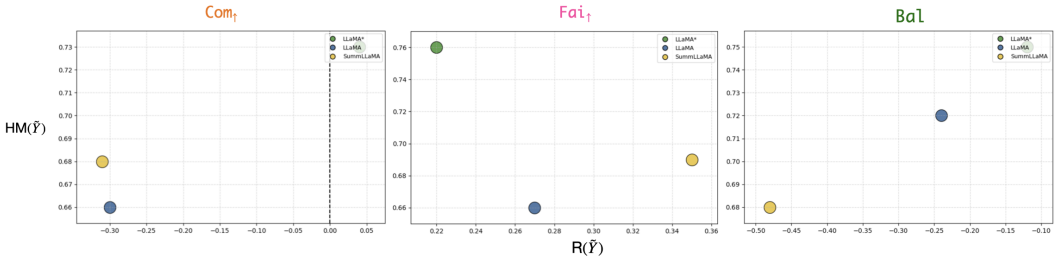

Figure 11: Model performance across different control settings. Each point represents the mean of HM($\tilde{Y}$) (y-axis) and R($\tilde{Y}$) (x-axis) across all test cases in FeedSum test set. The three panels show performance under different control priorities: Com$_\uparrow$ prioritizes completeness, Fai$_\uparrow$ prioritizes faithfulness, and Bal aims to balance both. The vertical dashed line at R($\tilde{Y}$) = 0 represents the controllability target (reference) between completeness and conciseness.

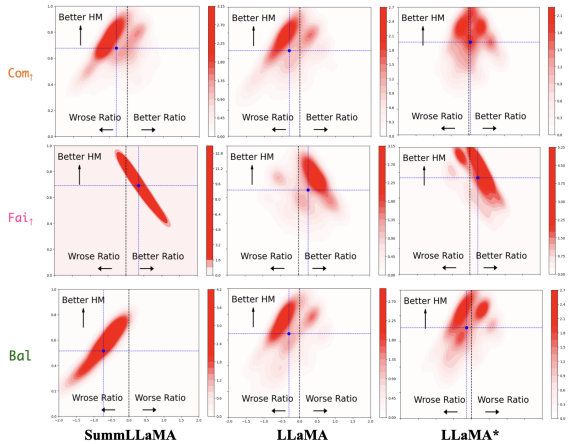

Figure 12: Distributions of R($\tilde{Y}$) (x) and HM($\tilde{Y}$) (y) metrics. Com$_\uparrow$ prioritizes completeness, Fai$_\uparrow$ prioritizes faithfulness, and Bal to balances them. Blue dashed lines mark the mean of the metrics, and the arrows point in the direction in which metrics are better or worse.

Model	Source	Criteria	S_{com}	S_{con}	HM($\tilde{Y}$)	R($\tilde{Y}$)	Succ rate	Prop. of 1
		$Com_\uparrow$	0.70 ± 0.23	0.59 ± 0.25	0.61 ± 0.19	0.22 ± 0.47	0.82	0.13
	dialogsum	$Con_\uparrow$	0.62 ± 0.19	0.80 ± 0.25	0.66 ± 0.16	0.25 ± 0.5	0.72	0.22
		Bal	0.71 ± 0.24	0.74 ± 0.27	0.69 ± 0.22	-0.03 ± 0.44	0.94	-
		$Com_\uparrow$	0.65 ± 0.21	0.52 ± 0.22	0.55 ± 0.19	0.27 ± 0.48	0.94	0.00
LLaMA	cnn	$Con_\uparrow$	0.48 ± 0.18	0.76 ± 0.26	0.56 ± 0.18	0.47 ± 0.53	0.94	0.01
		Bal	0.55 ± 0.2	0.61 ± 0.25	0.55 ± 0.19	-0.10 ± 0.44	0.90	-
		$Com_\uparrow$	0.90 ± 0.21	0.42 ± 0.2	0.53 ± 0.15	0.82 ± 0.59	0.55	0.10
	wikihow	$Con_\uparrow$	0.75 ± 0.28	0.67 ± 0.26	0.62 ± 0.12	-0.09 ± 0.8	0.28	0.42
		Bal	0.91 ± 0.2	0.73 ± 0.28	0.76 ± 0.22	0.28 ± 0.57	0.68	-
		$Com_\uparrow$	0.63 ± 0.2	0.87 ± 0.22	0.69 ± 0.14	-0.34 ± 0.56	0.7	0.25
	dialogsum	$Con_\uparrow$	0.53 ± 0.16	0.98 ± 0.09	0.67 ± 0.14	0.67 ± 0.38	0.72	0.21
		Bal	0.69 ± 0.25	0.98 ± 0.1	0.78 ± 0.19	-0.41 ± 0.45	0.94	-
		$Com_\uparrow$	0.61 ± 0.2	0.62 ± 0.28	0.57 ± 0.18	0.04 ± 0.64	0.98	0.01
SummLLaMA	cnn	$Con_\uparrow$	0.4 ± 0.17	0.99 ± 0.05	0.55 ± 0.18	1.01 ± 0.49	0.96	0.01
		Bal	0.5 ± 0.19	0.94 ± 0.17	0.62 ± 0.18	-0.69 ± 0.5	0.97	-
		$Com_\uparrow$	0.68 ± 0.27	0.74 ± 0.28	0.61 ± 0.12	-0.09 ± 0.84	0.23	0.46
	wikihow	$Con_\uparrow$	0.48 ± 0.16	0.98 ± 0.09	0.62 ± 0.13	0.77 ± 0.43	0.15	0.58
		Bal	0.9 ± 0.21	1 ± 0.03	0.93 ± 0.15	-0.14 ± 0.34	0.74	-
		$Com_\uparrow$	0.71 ± 0.2	0.71 ± 0.25	0.67 ± 0.16	0.03 ± 0.5	0.7	0.24
	dialogsum	$Con_\uparrow$	0.59 ± 0.16	0.94 ± 0.15	0.7 ± 0.13	0.5 ± 0.4	0.62	0.32
		Bal	0.78 ± 0.23	0.89 ± 0.2	0.81 ± 0.19	-0.16 ± 0.42	0.96	-
		$Com_\uparrow$	0.67 ± 0.21	0.63 ± 0.24	0.62 ± 0.19	0.07 ± 0.45	0.96	0.03
LLaMA*	cnn	$Con_\uparrow$	0.47 ± 0.18	0.92 ± 0.19	0.59 ± 0.17	0.73 ± 0.51	0.96	0.02
		Bal	0.59 ± 0.21	0.78 ± 0.24	0.64 ± 0.19	-0.29 ± 0.46	0.99	-
		$Com_\uparrow$	0.86 ± 0.24	0.53 ± 0.22	0.6 ± 0.14	0.51 ± 0.66	0.46	0.31
	wikihow	$Con_\uparrow$	0.59 ± 0.26	0.85 ± 0.25	0.61 ± 0.13	0.43 ± 0.82	0.18	0.6
		Bal	0.92 ± 0.19	0.88 ± 0.23	0.86 ± 0.2	0.07 ± 0.49	0.79	-
		$Com_\uparrow$	0.75 ± 0.22	0.57 ± 0.22	0.62 ± 0.18	0.31 ± 0.46	0.8	0.14
	dialogsum	$Con_\uparrow$	0.65 ± 0.21	0.72 ± 0.24	0.65 ± 0.16	0.1 ± 0.47	0.76	0.18
		Bal	0.73 ± 0.24	0.7 ± 0.25	0.69 ± 0.21	0.06 ± 0.44	0.96	-
		$Com_\uparrow$	0.63 ± 0.19	0.52 ± 0.2	0.55 ± 0.17	0.22 ± 0.41	0.99	0
Qwen	cnn	$Con_\uparrow$	0.54 ± 0.19	0.62 ± 0.24	0.55 ± 0.18	0.1 ± 0.42	1	0
		Bal	0.58 ± 0.21	0.57 ± 0.22	0.56 ± 0.18	0.04 ± 0.4	1	-
		$Com_\uparrow$	0.91 ± 0.2	0.44 ± 0.19	0.56 ± 0.16	0.79 ± 0.55	0.64	0.1
	wikihow	$Con_\uparrow$	0.87 ± 0.23	0.52 ± 0.2	0.6 ± 0.12	-0.52 ± 0.58	0.38	0.3
		Bal	0.91 ± 0.21	0.67 ± 0.29	0.72 ± 0.22	0.37 ± 0.59	0.69	-
		$Com_\uparrow$	0.63 ± 0.21	0.74 ± 0.25	0.64 ± 0.16	-0.15 ± 0.55	0.68	0.28
	dialogsum	$Con_\uparrow$	0.57 ± 0.18	0.9 ± 0.19	0.67 ± 0.15	0.47 ± 0.42	0.7	0.24
		Bal	0.74 ± 0.23	0.84 ± 0.24	0.76 ± 0.21	-0.14 ± 0.4	0.94	-
		$Com_\uparrow$	0.58 ± 0.18	0.61 ± 0.23	0.57 ± 0.16	-0.01 ± 0.47	0.96	0.04
Qwen*	cnn	$Con_\uparrow$	0.48 ± 0.19	0.7 ± 0.25	0.54 ± 0.18	0.4 ± 0.51	0.97	0.02
		Bal	0.56 ± 0.2	0.65 ± 0.24	0.58 ± 0.18	-0.15 ± 0.45	1	-
		$Com_\uparrow$	0.86 ± 0.24	0.54 ± 0.23	0.6 ± 0.14	0.49 ± 0.71	0.45	0.27
	wikihow	$Con_\uparrow$	0.69 ± 0.26	0.75 ± 0.27	0.63 ± 0.12	0.09 ± 0.74	0.22	0.48
		Bal	0.9 ± 0.21	0.78 ± 0.26	0.8 ± 0.21	0.17 ± 0.53	0.67	-
		$Com_\uparrow$	0.73 ± 0.21	0.6 ± 0.22	0.63 ± 0.17	0.24 ± 0.46	0.81	0.14
	dialogsum	$Con_\uparrow$	0.63 ± 0.21	0.68 ± 0.26	0.61 ± 0.17	0.06 ± 0.53	0.71	0.22
		Bal	0.75 ± 0.24	0.72 ± 0.26	0.7 ± 0.22	0.05 ± 0.44	0.94	-
		$Com_\uparrow$	0.64 ± 0.2	0.52 ± 0.21	0.56 ± 0.18	0.24 ± 0.4	0.99	0
Mistral	cnn	$Con_\uparrow$	0.55 ± 0.19	0.63 ± 0.25	0.56 ± 0.18	0.12 ± 0.46	0.98	0.01
		Bal	0.59 ± 0.21	0.57 ± 0.22	0.56 ± 0.18	0.05 ± 0.43	1	-
		$Com_\uparrow$	0.9 ± 0.22	0.44 ± 0.2	0.55 ± 0.17	0.77 ± 0.61	0.56	0.11
	wikihow	$Con_\uparrow$	0.82 ± 0.26	0.56 ± 0.24	0.6 ± 0.14	-0.41 ± 0.7	0.41	0.3
		Bal	0.9 ± 0.21	0.63 ± 0.28	0.69 ± 0.22	0.43 ± 0.58	0.72	-
		$Com_\uparrow$	0.64 ± 0.21	0.79 ± 0.26	0.66 ± 0.17	-0.19 ± 0.56	0.66	0.27
	dialogsum	$Con_\uparrow$	0.58 ± 0.19	0.79 ± 0.26	0.63 ± 0.17	0.31 ± 0.53	0.68	0.28
		Bal	0.73 ± 0.24	0.8 ± 0.26	0.73 ± 0.21	-0.09 ± 0.45	0.92	-
		$Com_\uparrow$	0.62 ± 0.21	0.65 ± 0.26	0.6 ± 0.19	-0.02 ± 0.52	0.84	0.02
Mistral*	cnn	$Con_\uparrow$	0.49 ± 0.18	0.75 ± 0.26	0.56 ± 0.17	0.41 ± 0.54	0.92	0.01
		Bal	0.56 ± 0.21	0.64 ± 0.26	0.57 ± 0.2	-0.13 ± 0.44	0.93	-
		$Com_\uparrow$	0.84 ± 0.27	0.53 ± 0.24	0.58 ± 0.15	0.47 ± 0.79	0.39	0.33
	wikihow	$Con_\uparrow$	0.73 ± 0.28	0.65 ± 0.27	0.6 ± 0.13	-0.11 ± 0.78	0.28	0.46
		Bal	0.89 ± 0.22	0.78 ± 0.28	0.78 ± 0.22	0.17 ± 0.57	0.76	-
		$Com_\uparrow$	0.87 ± 0.19	0.55 ± 0.2	0.65 ± 0.18	0.5 ± 0.39	0.9	0.08
	dialogsum	$Con_\uparrow$	0.71 ± 0.2	0.73 ± 0.25	0.68 ± 0.16	0 ± 0.49	0.7	0.27
		Bal	0.83 ± 0.2	0.7 ± 0.24	0.73 ± 0.19	0.21 ± 0.41	0.96	-
		$Com_\uparrow$	0.74 ± 0.19	0.47 ± 0.18	0.56 ± 0.17	0.5 ± 0.44	1	0
GPT4omini	cnn	$Con_\uparrow$	0.59 ± 0.19	0.6 ± 0.23	0.57 ± 0.18	0 ± 0.43	0.99	0.01
		Bal	0.64 ± 0.2	0.55 ± 0.21	0.57 ± 0.18	0.19 ± 0.42	1	-
		$Com_\uparrow$	0.92 ± 0.2	0.41 ± 0.2	0.52 ± 0.17	0.88 ± 0.63	0.64	0.04
	wikihow	$Con_\uparrow$	0.83 ± 0.27	0.59 ± 0.24	0.61 ± 0.13	-0.35 ± 0.73	0.38	0.33
		Bal	0.92 ± 0.19	0.6 ± 0.29	0.67 ± 0.22	0.52 ± 0.64	0.72	-
		$Com_\uparrow$	0.84 ± 0.19	0.55 ± 0.2	0.64 ± 0.17	0.49 ± 0.42	0.9	0.1
	dialogsum	$Con_\uparrow$	0.69 ± 0.21	0.7 ± 0.24	0.65 ± 0.15	0 ± 0.5	0.72	0.24
		Bal	0.81 ± 0.21	0.72 ± 0.25	0.73 ± 0.21	0.15 ± 0.42	0.97	-
		$Com_\uparrow$	0.75 ± 0.19	0.47 ± 0.19	0.56 ± 0.17	0.52 ± 0.44	0.98	0.02
GPT4o	cnn	$Con_\uparrow$	0.58 ± 0.18	0.58 ± 0.22	0.55 ± 0.17	0 ± 0.39	0.98	0.01
		Bal	0.64 ± 0.2	0.55 ± 0.23	0.57 ± 0.18	0.21 ± 0.47	1	-
		$Com_\uparrow$	0.92 ± 0.19	0.39 ± 0.19	0.51 ± 0.18	0.94 ± 0.55	0.67	0.06
	wikihow	$Con_\uparrow$	0.83 ± 0.26	0.55 ± 0.24	0.59 ± 0.13	-0.42 ± 0.71	0.37	0.32
		Bal	0.92 ± 0.2	0.61 ± 0.29	0.68 ± 0.23	0.51 ± 0.63	0.7	-
		$Com_\uparrow$	0.92 ± 0.15	0.67 ± 0.19	0.76 ± 0.14	0.34 ± 0.36	0.7	0.3
	dialogsum	$Con_\uparrow$	0.74 ± 0.16	0.85 ± 0.2	0.77 ± 0.12	0.13 ± 0.36	0.38	0.6
		Bal	0.92 ± 0.16	0.88 ± 0.17	0.89 ± 0.14	0.04 ± 0.27	1	-
		$Com_\uparrow$	0.83 ± 0.16	0.61 ± 0.18	0.68 ± 0.14	0.33 ± 0.4	0.97	0.03
GPT4-turbo	cnn	$Con_\uparrow$	0.65 ± 0.18	0.82 ± 0.2	0.7 ± 0.15	0.24 ± 0.43	0.96	0.04
		Bal	0.74 ± 0.17	0.75 ± 0.2	0.72 ± 0.15	0 ± 0.35	1	-
		$Com_\uparrow$	0.93 ± 0.18	0.51 ± 0.2	0.62 ± 0.16	0.64 ± 0.53	0.65	0.22
	wikihow	$Con_\uparrow$	0.75 ± 0.28	0.74 ± 0.25	0.66 ± 0.11	0.02 ± 0.76	0.25	0.62
		Bal	0.94 ± 0.17	0.8 ± 0.25	0.83 ± 0.19	0.19 ± 0.48	0.86	-
		$Com_\uparrow$	0.89 ± 0.18	0.52 ± 0.19	0.63 ± 0.18	0.61 ± 0.39	0.94	0.04
	dialogsum	$Con_\uparrow$	0.64 ± 0.21	0.72 ± 0.25	0.64 ± 0.16	0.09 ± 0.5	0.64	0.31
		Bal	0.8 ± 0.22	0.71 ± 0.25	0.73 ± 0.21	0.14 ± 0.41	0.96	-
		$Com_\uparrow$	0.78 ± 0.18	0.43 ± 0.19	0.53 ± 0.17	0.67 ± 0.47	1	0
Gemini	cnn	$Con_\uparrow$	0.5 ± 0.18	0.67 ± 0.25	0.55 ± 0.17	0.27 ± 0.46	0.96	0.02
		Bal	0.61 ± 0.19	0.62 ± 0.23	0.59 ± 0.17	0.01 ± 0.42	0.98	-
		$Com_\uparrow$	0.92 ± 0.2	0.39 ± 0.17	0.52 ± 0.17	0.91 ± 0.59	0.64	0.06
	wikihow	$Con_\uparrow$	0.82 ± 0.26	0.55 ± 0.22	0.6 ± 0.11	-0.4 ± 0.67	0.4	0.34
		Bal	0.91 ± 0.21	0.66 ± 0.29	0.71 ± 0.22	0.39 ± 0.59	0.7	-

Table 14: Detailed comparison of model performance using FineSurE. The value shows averages over all cases across two in domain datasets with standard deviations. $*$ denotes foundation models fine-tuned using our method. $Com_\uparrow$ emphasizes completeness, $Con_\uparrow$ emphasizes conciseness, and Bal seeks to balance both. *Succ rate* refers to the proportion of cases where either S_{com} or S_{con} is non-zero. *Proportion of 1* indicates the proportion of cases where both S_{com} and S_{con} equal 1. In the $Com_\uparrow$ and $Con_\uparrow$ settings, such cases are considered uncontrollable and are excluded from the computation of S_{com}, S_{con}, HM($\tilde{Y}$), and R($\tilde{Y}$). In contrast, under the Bal setting, these cases are treated as ideal (i.e., perfect) and are retained (denoted as "–").

Model	Criteria	$\mathbf{S_{com}}$	$\mathbf{S_{fai}}$	$\mathbf{HM}(\tilde{Y})$	$\mathbf{R}(\tilde{Y})$	Succ rate	Prop. of 1
LLaMA	$\text{Com}_{\uparrow}$	0.63±0.24	0.80±0.20	0.66±0.18	-0.30±0.56	0.64	0.20
	$\text{Fai}_{\uparrow}$	0.62±0.22	0.80±0.23	0.66±0.16	0.27±0.50	0.63	0.24
	Bal	0.69±0.27	0.82±0.21	0.72±0.21	-0.24±0.45	0.87	-
SummLLaMA	$\text{Com}_{\uparrow}$	0.63±0.21	0.83±0.20	0.68±0.16	-0.31±0.50	0.65	0.24
	$\text{Fai}_{\uparrow}$	0.61±0.18	0.84±0.07	0.69±0.13	0.35±0.35	0.62	0.26
	Bal	0.61±0.31	0.86±0.16	0.68±0.27	-0.48±0.63	0.89	-
LLaMA*	$\text{Com}_{\uparrow}$	0.77±0.23	0.73±0.19	0.73±0.16	0.04±0.39	0.79	0.09
	$\text{Fai}_{\uparrow}$	0.72±0.24	0.86±0.15	0.76±0.16	0.22±0.38	0.70	0.18
	Bal	0.75±0.25	0.81±0.18	0.75±0.19	-0.12±0.40	0.89	-

Table 15: Detailed comparison of model performance using FineSurE. The value shows averages over all cases in FeedSum with standard deviations. $*$ indicates the foundation model fine-tuned using our method. $\text{Com}_{\uparrow}$ prioritizes faithfulness, $\text{Fai}_{\uparrow}$ prioritizes faithfulness, and Bal aims to balance both. *Succ rate* denotes the proportion of cases where either S_{com} or S_{fai} is non-zero. *Proportion of 1* indicates the proportion of cases where both S_{com} and S_{fai} equal 1. In the $\text{Com}_{\uparrow}$ and $\text{Fai}_{\uparrow}$ settings, these are considered uncontrollable cases and are excluded when computing S_{com}, S_{con}, $\text{HM}(\tilde{Y})$, and $\text{R}(\tilde{Y})$. In contrast, under Bal, these cases are treated as ideal (*i.e.*, perfect), so they are retained (denoted as "–").