FlexTag: Instruction Tagging for Multi-Perspective Understanding with Small Language Models

Anonymous ACL submission

Abstract

Instruction tagging maps free-form user requests into structured metadata and serves as a fundamental component for LLM routing, data auditing, and retrieval indexing. Existing tagging methods typically rely on flat semantic keyword generation, which struggles to generalize across template variations and fails to capture multi-perspective tag quality. In practice, high-quality tags should exhibit diversity, complementarity, and synergy across multiple semantic facets. We introduce FlexTag, a 1B-scale instruction tagging framework that supports both open-ended multi-perspective tag generation and tag selection. FlexTag addresses limitations of prior approaches through three key components: (1) a multi-template data augmentation pipeline that improves robustness to prompt and template variations, (2) a multi-objective training strategy that optimizes for multi-perspective tag quality, and (3) a template-adaptive inference pipeline that enhances tagging reliability. Experiments on four public datasets demonstrate that FlexTag outperforms competitive small-model baselines in open tag generation, surpasses GPT-4o in zero-shot tag selection, and maintains strong generalization under out-of-distribution settings. We release FLEXTAG at https://anonymous.4open.science/r/FlexTag-A0DD.

1 Introduction

In recent years, large language models (LLMs) such as GPT-4 (Achiam et al., 2023) have become core infrastructure for real-world systems, powering code generation, question answering, and writing assistance. A key enabling technique behind these applications is instruction tagging, which maps free-form natural language requests into structured metadata labels. Such tags support a broad range of downstream uses: LLM routing can select an appropriately sized model to balance quality and cost; data analysis can audit training distributions and reveal coverage gaps; and retrieval can build fine-grained indices that improve search and navigation.

Most prior work on tag generation (*eg.,* InsTagger (Lu et al., 2023), TAGROUTER (Chen et al., 2025)) targets open-domain keyword tagging: it leverages strong LLMs to produce arbitrary keyword tags and then distills this behavior into smaller models for large-scale analysis. TagGPT (Li et al., 2023) instead performs zero-shot tag generation with general-purpose LLMs, using frequency statistics and semantic matching to induce tag sets. In a different direction, TagLLM (Shen et al., 2024) treats tags as controllable inputs via learnable embeddings to steer model behavior, rather than generating human-readable, structured metadata.

While large models such as GPT-4o (Li et al., 2023) can perform instruction tagging via zero-shot prompting, production systems are highly sensitive to latency, cost, and privacy, making 1B-scale small language models a more practical deployment choice. However, the limited capacity of small models amplifies three deeply interconnected limitations of existing approaches. First, existing methods are designed primarily for open-domain keyword tagging and cannot meet production demands for schema-constrained multi-facet annotations that directly drive routing, auditing, and retrieval pipelines. Second, even when output format is enforced, standard fine-tuning under low-resource conditions fails to guarantee multi-perspective quality: models collapse toward high-frequency label values, produce redundant predictions across facets, and fail to generate complementary evidence across dimensions. Third, small models trained under narrow task-specific supervision struggle to retain broad instruction-following competence, causing sharp degradation under minor changes in input format or output schema. Together, these challenges constitute the core obstacles to building high-quality small-scale taggers, and must be addressed jointly across data, training,

and inference.

To address the above challenges, we propose FLEXTAG, a 1B-scale instruction tagging framework supporting both tag selection and open-ended multi-perspective tag generation, tackling them jointly from three complementary angles: *(1)* a low-resource multi-template data augmentation pipeline that rewrites each sample into four template categories to mitigate template rigidity; *(2)* a multi-objective training strategy with span-weighted cross-entropy, facet coverage, cross-facet redundancy, and synergy terms to ensure multi-perspective quality; and *(3)* a template-adaptive inference pipeline to guarantee output validity. Our main contributions are as follows:

- We propose a low-resource multi-template data augmentation pipeline that augument each sample into four instruction formats, systematically mitigating template rigidity for small-scale taggers.
- We introduce a multi-objective training strategy comprising span-weighted cross-entropy, facet coverage, cross-facet redundancy, and synergy, jointly ensuring multi-perspective annotation quality under low-resource conditions.
- Extensive experiments across four public datasets demonstrate that FLEXTAG surpasses all comparable-scale baselines on open generation, outperforms GPT-4o zero-shot on tag selection, and maintains strong generalization on out-of-distribution datasets.

2 Related Work

Instruction Tagging. Instruction tagging aims to map free-form user requests into structured metadata. INSTAG (Lu et al., 2023) leverages ChatGPT to generate open-ended keyword tags for analyzing SFT data diversity and complexity. TAGROUTER (Chen et al., 2025) uses tags for LLM routing to balance cost and performance. TAGGPT (Li et al., 2023) explores zero-shot multi-modal tagging via prompt engineering. These methods focus on flat keyword generation and do not support multi-dimensional schema-constrained annotations or multi-perspective quality guarantees. Beyond instruction tagging, TAGLLM (Shen et al., 2024) repurposes tags as learnable embeddings to control model behavior, and LLM4TAG (Tang et al., 2025) targets industrial-scale content tagging via graph-based recall and LLM-based selection, both representing orthogonal directions to structured metadata annotation. Work on instruction tuning (Wei et al., 2022; Wang et al., 2022, 2023) and data augmentation (Xu et al., 2025) further demonstrates that diverse supervision signals are critical for generalization. FLEXTAG builds on these directions and targets multi-perspective tagging supporting both tag selection and open generation.

Small Language Models. Language models are highly sensitive to prompt wording and formatting, where semantically equivalent variants can cause substantial performance fluctuations (Razavi et al., 2025; Sclar et al., 2024; Zhuo et al., 2024). This challenge is particularly acute for SLMs, which are attractive for deployment due to low latency and cost (Ding et al., 2023; Gunasekar et al., 2023) but struggle to retain broad instruction-following competence under narrow task-specific supervision (Gudibande et al., 2023). In schema-constrained multi-template tagging, this limitation is further amplified: the model must maintain stable outputs across diverse instruction formats while satisfying multi-perspective quality requirements under low-resource conditions. Template-based augmentation (Longpre et al., 2023; Mishra et al., 2022) is a common mitigation strategy, and post-training methods such as SFT, DPO (Rafailov et al., 2023), and GRPO (Shao et al., 2024) expand the adaptation toolbox. However, optimizing SLMs for structured multi-template tagging remains underexplored. FLEXTAG addresses this gap by combining multi-template augmentation with structure-aware training objectives.

3 Preliminaries

Notations. Let $\mathcal{X}$ be the space of natural-language queries and $\mathcal{T} = \{\tau_1, \ldots, \tau_K\}$ the set of tagging templates. Each template τ defines an input/output protocol, inducing an output space $\mathcal{Y}_\tau$ and a valid subset $\mathcal{Y}_\tau^{\text{val}} \subseteq \mathcal{Y}_\tau$ satisfying template-specific format and value constraints. Each valid output is a list of perspective-aware tag objects,

$$y = \{(p_j, t_j, e_j)\}_{j=1}^{m},$$

where p_j is a dynamically selected perspective, t_j is its semantic tag, and e_j is a brief explanation. Both the number of items and the set of perspectives are instruction-dependent. For tag-selection tem-

plates, t_j is selected from a candidate set; for open-generation templates, t_j is freely generated under the same JSON schema. The dataset is grouped as $\mathcal{D} = \bigcup_{\tau \in \mathcal{T}} \mathcal{D}_\tau$, where $\mathcal{D}_\tau = \{(x_i, y_i)\}_{i=1}^{N_\tau}$ and $y_i \in \mathcal{Y}_\tau^{\text{val}}$.

Tagging objective. Given (x, τ), the tagger predicts $\hat{y} = f_\theta(x; \tau) \in \mathcal{Y}_\tau$. We optimize

$$\min_\theta \; \mathbb{E}_{\tau \sim p(\tau)} \left[\frac{1}{|\mathcal{D}_\tau|} \sum_{(x_i, y_i) \in \mathcal{D}_\tau} \mathcal{L}_{\text{NLL}}(f_\theta(x_i; \tau), y_i) \right], \tag{1}$$

where $p(\tau)$ is the template sampling distribution. This likelihood objective alone does not explicitly enforce template validity or multi-perspective quality, such as avoiding perspective collapse and redundant tags.

4 Method

Figure 1 gives an overview of FLEXTAG, which addresses the three challenges through multi-template data augmentation, multi-objective training, and template-adaptive inference.

4.1 Multi-template Data Augumentation

Existing instruction tagging datasets typically cover a single template, causing models to entangle the annotation schema with a fixed input format and degrade under template shifts. We therefore build a multi-template augmentation pipeline that projects the same semantic supervision signal across heterogeneous template formats (Details in Appendix C). The pipeline has three steps. *(i)* We aggregate raw samples from multiple public instruction sources and prompt GPT-4o to produce multi-facet structured labels, yielding a base annotation set $\mathcal{D}_{\text{base}}$ that serves as the semantic anchor for template projection. *(ii)* We filter $\mathcal{D}_{\text{base}}$ by discarding samples with extremely low-frequency tag values and removing samples with missing facet fields. *(iii)* We apply multi-template rewriting to each sample in $\mathcal{D}_{\text{base}}$, producing a set of template instances $\{(x^{(\tau)}, y^{(\tau)})\}_{\tau \in \mathcal{T}(x)}$. The rewriting covers four template categories (open-tag generation, tag selection, zero-shot, and few-shot), each with multiple variants, and we verify the projected supervision satisfies $y^{(\tau)} \in \mathcal{Y}_\tau^{\text{val}}$. The resulting dataset is grouped by template as $\mathcal{D} = \bigcup_{\tau \in \mathcal{T}} \mathcal{D}_\tau$.

4.2 Tag-focused Supervision

In multi-template tagging, model outputs typically consist of structured tag fields interleaved with auxiliary text such as explanations or structural tokens. Applying uniform token-level NLL across all positions dilutes the training signal with low-information-density spans, an effect that becomes more pronounced when templates contain lengthy auxiliary text. We therefore propose a span-weighted supervision strategy that assigns an amplified weight $\alpha > 1$ to tag-value spans and weight 1 to all remaining tokens:

$$\mathcal{L}_{\text{wNLL}} = -\sum_{t=1}^{|y|} m_t \log p_\theta(y_t \mid y_{<t}, x, \tau), \tag{2}$$

where $m_t = \alpha$ if position t falls within a tag-value span and $m_t = 1$ otherwise. This weighting relies solely on the parseable boundaries of tag fields, requiring no facet-specific semantic knowledge, and thus applies uniformly across all template variants. Intuitively, $\mathcal{L}_{\text{wNLL}}$ redistributes the gradient budget away from verbose auxiliary text toward high-information-density label spans, improving tag field prediction quality across templates.

4.3 Multi-perspective Generation

While $\mathcal{L}_{\text{wNLL}}$ improves supervision on tag fields, token-level likelihood training alone leaves three desiderata unaddressed for perspective-aware structured generation templates $\mathcal{T}_{\text{gen}} \subseteq \mathcal{T}$. Here, we use *facet* to refer to a dynamically selected perspective in the output schema: each generated item contains a facet, a tag, and a brief explanation. The desiderata are: *(i) coverage*—generated facet–tag pairs should preserve diverse semantic coverage rather than collapsing toward high-frequency labels; *(ii) de-redundancy*—different facets should not degenerate into redundant restatements of the same semantic signal; *(iii) synergy*—facets should provide complementary evidence such that jointly generated facets capture richer intent information than isolated facet predictions. Inspired by partial information decomposition (PID) theory (Venkatesh et al., 2023), we introduce three lightweight regularizers targeting these goals.

Facet Coverage. For each frequent facet p under template τ, we compute an empirical tag prior $\pi_{\tau,p}$ over observed tags in the training annotations. Let $\hat{p}_{\tau,p}$ be the batch-averaged model-induced distribution over the same tracked tags. We penalize

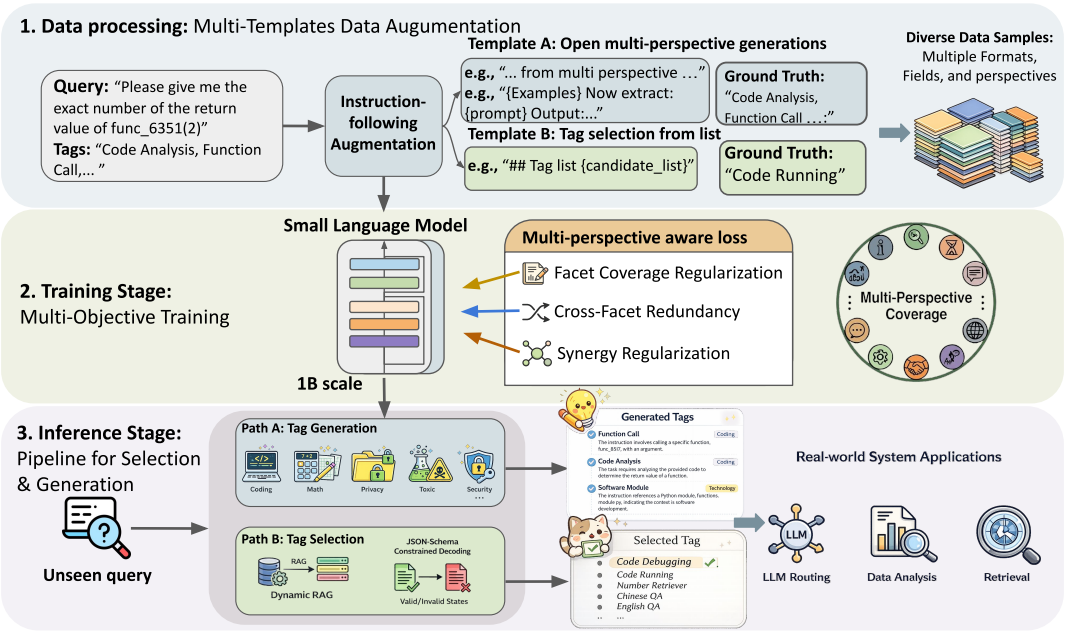

Figure 1: **FlexTag** overview. Semantically aligned multi-template augmentation yields diverse tagging templates. A single 1B tagger is trained with span-weighted supervision and multi-perspective regularizers (coverage, de-redundancy, synergy). At inference, open-generation templates decode multi-perspective structured tags, while selection templates use RAG few-shot demonstrations and template-derived constrained decoding to ensure validity.

collapse toward high-frequency predictions by

$$\mathcal{L}_{\text{cover}} = \frac{1}{|\mathcal{P}_\tau|} \sum_{p \in \mathcal{P}_\tau} \mathrm{KL}(\pi_{\tau,p} \,\|\, \hat{p}_{\tau,p}), \qquad (3)$$

where $\mathcal{P}_\tau$ is the set of tracked frequent facets. Anchoring to empirical priors preserves natural data skew while discouraging excessive collapse.

Cross-Facet Redundancy. Even with adequate coverage, the model may generate multiple facet–tag pairs that express nearly identical semantic content. Such redundancy weakens the intended multi-perspective structure, since different facets should capture complementary aspects of the instruction rather than restating the same signal. For each generated facet–tag item j in sample x, we define a span-level predictive representation by averaging the model's token distributions over the positions assigned to that item:

$$\hat{p}_j(x) = \frac{1}{|\mathcal{P}_j(x)|} \sum_{t \in \mathcal{P}_j(x)} p_\theta(\cdot \mid x, y_{<t}, \tau), \qquad (4)$$

where $\mathcal{P}_j(x)$ denotes the token positions corresponding to the tag-value span of item j, grouped by its facet. For any pair of distinct generated items $j \neq j'$, we compute their distributional similarity:

$$\rho_{j,j'}(x) = \cos\big(\hat{p}_j(x),\, \hat{p}_{j'}(x)\big). \qquad (5)$$

We penalize positive similarity among distinct facet–tag items:

$$\mathcal{L}_{\text{redund}} = \sum_{x:\, \tau(x) \in \mathcal{T}_{\text{gen}}} \sum_{j<j'} \max(0,\ \rho_{j,j'}(x)). \qquad (6)$$

This soft constraint discourages semantic co-movement among generated facet–tag pairs without imposing strict statistical independence.

Synergy. To operationalize synergy, we encourage different generated facet–tag items to provide complementary information. For each training sample, let

$$\mathcal{A}(y) = \{a_1, \ldots, a_m\}$$

denote the set of facet–tag items in the structured output, where each item $a_j = (p_j, t_j)$ consists of a dynamically selected facet and its associated tag. At each training step, we sample a target item $s \sim \mathrm{Unif}(\mathcal{A}(y))$ and a source item $k \sim \mathrm{Unif}(\mathcal{A}(y) \setminus \{s\})$. Let $\mathcal{L}_{\text{anchor}}^{(\text{all}\to s)}$ denote the span-level NLL for predicting the target item s when all other facet–tag items are available as context, and let $\mathcal{L}_{\text{anchor}}^{(k\to s)}$ denote the corresponding loss when only the source item k is retained as context. We impose the following margin constraint:

$$\begin{aligned} \Delta_{s,k} &= \mathrm{sg}(\mathcal{L}_{\text{anchor}}^{(k\to s)}) - \mathcal{L}_{\text{anchor}}^{(\text{all}\to s)}, \\ \mathcal{L}_{\text{syn}} &= \mathbb{E}_{s,k}\big[\max(0,\ \gamma - \Delta_{s,k})\big]. \end{aligned} \qquad (7)$$

Algorithm 1 Template-adaptive constrained inference for selection templates.

Require: Input instruction x, selection template $\tau \in \mathcal{T}_{\text{sel}}$, example library $\mathcal{B}_\tau$, tagger f_θ, $\mathbf{CD}(f_\theta, \mathbf{P}, \mathcal{C})$: constrained decoding of f_θ under prompt $\mathbf{P}$ and constraints $\mathcal{C}$
Ensure: Structured prediction $\hat{y} \in \mathcal{Y}_\tau^{\text{val}}$
1: Retrieve demonstrations $\mathcal{E}_\tau(x)$ from $\mathcal{B}_\tau$
2: Construct $\mathbf{P} \leftarrow \text{Prompt}(x, \tau, \mathcal{E}_\tau(x))$
3: Derive $\mathcal{C}_\tau$ from template rules (structure + admissible candidates)
4: $\hat{y} \leftarrow \mathbf{CD}(f_\theta, \mathbf{P}, \mathcal{C}_\tau)$
5: **return** $\hat{y}$

where $\gamma > 0$ is a margin hyperparameter and $\text{sg}(\cdot)$ denotes stop-gradient.

4.4 Objective Overview

We optimize a tag-focused likelihood term for all templates, and additionally apply multi-perspective regularizers for perspective-aware generation templates $\tau \in \mathcal{T}_{\text{gen}}$. The per-sample objective is:

$$\begin{aligned} \mathcal{L}(x, y; \tau) &= \mathcal{L}_{\text{wNLL}}(x, y; \tau) + \mathbf{1}_{\tau \in \mathcal{T}_{\text{gen}}} \lambda \mathcal{R}(x, y; \tau), \\ \mathcal{R}(x, y; \tau) &= \mathcal{L}_{\text{cover}} + \mathcal{L}_{\text{redund}} + \mathcal{L}_{\text{syn}}, \end{aligned} \tag{8}$$

where $\lambda \geq 0$ (*e.g.,* 0.01) controls the overall strength of the multi-perspective regularization terms.

4.5 Template-Adaptive Constrained Inference

While training-time objectives improve tag prediction quality, a small language model can remain brittle at inference time under template shifts. This is most pronounced for *selection-style* templates, where the output must satisfy strict structure and candidate-set constraints (i.e., $\hat{y} \notin \mathcal{Y}_\tau^{\text{val}}$ leads to invalid predictions). We therefore design a template-adaptive inference procedure specifically for $\tau \in \mathcal{T}_{\text{sel}}$ that combines retrieval-augmented few-shot prompting with constrained decoding (Geng et al., 2023; Lewis et al., 2020) to ensure output validity. For each selection template τ, we maintain a demonstration library $\mathcal{B}_\tau$ of template-formatted input–output examples. Given (x, τ), we retrieve a small set of relevant demonstrations from $\mathcal{B}_\tau$ and prepend them to the prompt as few-shot context, improving template following and reducing formatting errors. We additionally derive a constraint set $\mathcal{C}_\tau$ from template rules, capturing structural constraints (field/layout) and value constraints (admissible candidate sets). During decoding, we mask invalid continuations that violate $\mathcal{C}_\tau$, guaranteeing $\hat{y} \in \mathcal{Y}_\tau^{\text{val}}$ for selection templates. Algorithm 1 summarizes the inference pipeline.

5 Experiment

5.1 Settings

Training Data. We construct FlexTag-Bench as our training corpus, aggregated from 18 publicly available instruction sources, and annotated by GPT-4o with subsequent quality filtering. The final dataset comprises approximately 28K samples for supervised fine-tuning and 343 samples for preference optimization. Details of the data construction pipeline and statistics are provided in Appendix B.

Evaluation Datasets. To ensure fair evaluation and validate generalization, we evaluate on four public datasets, using GPT-4o to generate ground-truth multi-perspective tag annotations under a unified protocol (Appendix D). `InfiniteBench` (Zhang et al., 2024) is a long-context benchmark over 12 domains (50 instances each, 600 total). `Helmet` (Yen et al., 2024) covers 7 task categories including retrieval-augmented generation, summarization, and long-document QA (100 each, 700 total). `RouterBench` (Hu et al., 2024) aggregates prompts from MBPP (Austin et al., 2021), GSM-8k (Cobbe et al., 2021), MMLU (Hendrycks et al., 2020), and MT-Bench (Bai et al., 2024), covering code generation, mathematical reasoning, and commonsense QA (1K). `OASST1` (Köpf et al., 2023) is a crowd-sourced multilingual conversation corpus over 35 languages (1K). Since FlexTag-Bench contains no long-context or long-document data, we treat `InfiniteBench` and `Helmet` as source-held-out OOD benchmarks; `RouterBench` and `OASST1` serve as in-domain held-out benchmarks, drawing training and evaluation samples from their respective train/test splits to ensure split-level separation without sample-level contamination. For inputs exceeding the context window, we retain the first and last 300 tokens.

Baselines. We select baselines along four dimensions for a comprehensive comparison: *(1) Task-specialized taggers*: INSTAGGER (Lu et al., 2023)[7B] and TAGROUTER (Chen et al., 2025)[0.5B], representing mainstream approaches specifically fine-tuned for instruction tagging. *(2) General-purpose small language models*: MISTRAL-v0.1-Instruct[7B] (Jiang et al., 2023), PHI-3.5-vision-instruct[4.2B] and PHI-3.5-mini-instruct[3.8B] (Abdin et al., 2024), LLAMA-3.2-Instruct[3B] (Grattafiori et al., 2024), GEMMA3 (Team et al., 2025) [1B],

Table 1: Open-generation tagging results on `InfiniteBench`, `Helmet`, `RouterBench`, and `OASST1`. We compare against prior tagging-specific methods and pretrained small language models. We report semantic accuracy (Recall, Precision, and F1) in %. The best results are in **bold**, and the second-best results are underlined.

Method	InfiniteBench			Helmet			RouterBench			OASST1		
	Recall	**Precision**	**F1**	**Recall**	**Precision**	**F1**	**Recall**	**Precision**	**F1**	**Recall**	**Precision**	**F1**
Mistral-v0.1-Instruct `[7B]`	47.6	31.4	34.9	32.8	20.5	22.4	45.8	22.7	27.4	62.7	40.1	46.3
Phi-3.5-vision-instruct `[4.2B]`	44.9	32.5	34.8	24.4	16.9	18.2	19.8	14.5	15.3	55.3	41.8	44.9
Phi-3.5-mini-instruct`[3.8B]`	**58.3**	40.6	46.2	25.5	18.4	19.7	23.2	17.3	18.7	**70.5**	50.3	56.4
Llama-3.2-Instruct `[3B]`	51.4	37.0	40.0	31.9	23.1	24.9	**68.8**	50.1	55.3	49.3	33.3	37.9
Gemma3 `[1B]`	37.0	37.3	33.8	16.9	16.1	14.8	35.9	30.3	30.2	60.3	46.4	49.9
Qwen3.5 `[0.8B]`	41.4	44.8	41.3	27.2	30.5	27.1	38.2	43.2	35.5	37.3	34.9	34.4
Qwen3 `[0.6B]`	40.6	50.0	41.2	18.5	27.6	20.3	27.1	37.9	30.0	48.4	52.8	47.9
Qwen2.5-Instruct `[0.5B]`	11.7	15.1	12.1	7.1	15.4	8.9	3.4	6.4	4.0	30.6	35.8	29.7
Instagger `[7B]`	50.0	**55.1**	49.3	40.5	38.1	37.2	49.7	55.4	50.2	38.6	48.2	39.9
TagRouter `[0.5B]`	27.3	36.5	30.3	21.4	24.4	21.8	24.0	21.0	21.4	18.8	23.4	19.9
FlexTag w/ DPO `[1B]`	53.6	51.8	**51.4**	**46.8**	**43.1**	**43.0**	59.8	**61.1**	**58.9**	60.7	**61.6**	**59.3**
FlexTag w/ GRPO `[1B]`	51.5	50.3	49.8	46.0	42.9	42.9	58.7	60.6	58.5	60.3	61.4	59.0
FlexTag (SFT only) `[1B]`	51.3	50.2	49.6	45.6	41.7	41.6	58.7	60.4	58.0	60.0	61.3	58.7

QWEN3.5 [0.8B], QWEN3[0.6B], and QWEN2.5-Instruct[0.5B] (Yang et al., 2025), evaluated in a zero-shot setting. These models are comparable in scale to FLEXTAG and serve to validate the necessity of task-specific training. *(3) Skyline*: GPT-4O (Achiam et al., 2023) and GPT-4O w/ RAG represent the general performance ceiling. For tag selection, GPT-4O w/ RAG incorporates retrieval-augmented few-shot demonstrations. All baselines are evaluated using identical inference prompts to ensure fair comparison.

Evaluation Metrics. We consider two evaluation settings: open-ended tag generation and candidate-constrained tag selection.

(i) Open Generation. Since exact string matching is overly strict and cannot accommodate benign lexical variations, we adopt semantic soft matching based on PHRASEBERT (Wang et al., 2021): a ground-truth tag is considered matched if its cosine similarity with any predicted tag exceeds the threshold $\sigma = 0.7$. Based on this rule, we compute sample-level Recall, Precision, and F1, and report macro-averaged scores over the test set. Beyond semantic correctness, we further evaluate whether predicted tags satisfy the quality requirements of multi-facet tagging through two additional metrics. Coverage measures whether the predicted tag distribution aligns with the ground-truth distribution rather than collapsing to a few dominant values:

$$\text{Coverage} = 1 - \frac{\text{JSD}(\hat{p} \| p^*)}{\log 2} \tag{9}$$

where $\hat{p}$ and p^* denote the predicted and ground-truth distributions, respectively, and JSD (Lin, 2002) is normalized to $[0, \log 2]$. Higher Coverage indicates better distributional alignment. Redundancy measures the degree of semantic overlap among predicted tags:

$$\text{Redundancy} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{2}{k_s(k_s - 1)} \sum_{i<j} \cos(\mathbf{e}_i, \mathbf{e}_j) \tag{10}$$

where $\mathcal{S}$ is the set of samples with $k_s \geq 2$ predicted tags, and $\mathbf{e}_i$ denotes the PHRASEBERT (Wang et al., 2021) embedding of tag t_i. Lower redundancy indicates that the model generates complementary rather than repetitive labels.

(ii) Tag Selection. We use Exact Match (EM), which measures the proportion of samples whose predicted labels exactly match the ground-truth labels:

$$\text{EM} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[\hat{y}_i = y_i] \tag{11}$$

where $\hat{y}_i$ and y_i denote the predicted and ground-truth labels for the i-th sample, respectively.

5.2 Main Results

Open Tagging Generation Performance. Table 1 presents open-generation results across four datasets. FLEXTAG at 1B scale outperforms nearly all baselines across all four datasets. On RouterBench, FLEXTAG w/ GRPO (Shao et al., 2024) achieves 58.5% F1, while the same-scale GEMMA3 [1B] obtains only 30.2% F1. On two OOD test sets, FLEXTAG maintains a substantial advantage with 51.4% F1 on InfiniteBench and 43.0% F1 on HELMET. Notably, task-specialized taggers still trail FLEXTAG on open generation: despite having 7×

Table 2: Tag selection performance (Exact Match, %) on `InfiniteBench` and `Helmet`. Higher is better. The best results are in **bold**, and the second-best results are underlined.

Method	InfiniteBench	Helmet
GPT-4o w/ RAG [>100B]	**91.6**	83.7
GPT-4o [>100B]	82.3	52.6
Mistral-v0.1-Instruct [7B]	74.2	53.9
Phi-3.5-vision-instruct [4.2B]	75.0	23.1
Phi-3.5-mini-instruct[3.8B]	74.3	54.8
Llama-3.2-Instruct [3B]	52.7	67.0
Gemma3 [1B]	33.3	24.3
Qwen3.5 [0.8B]	30.0	8.7
Qwen3 [0.6B]	24.1	37.4
Qwen2.5-Instruct [0.5B]	12.5	9.3
Instagger [7B]	6.9	0*
TagRouter [0.5B]	0*	0*
FlexTag w/ DPO [1B]	85.2	**84.6**
FlexTag w/ GRPO [1B]	85.2	82.3
FlexTag (SFT only) [1B]	85.4	82.4

*: A value of 0 indicates that the model produced no correct predictions under the corresponding evaluation metric.

more parameters, INSTAGGER [7B] stays below FLEXTAG across all datasets (*e.g.*, 49.3 vs. 51.4% F1 on InfiniteBench), and TAGROUTER [0.5B] remains in the 20–30% F1 range, as these methods are designed for coarse-grained keyword tagging and fail to capture multi-perspective quality requirements. General-purpose small models including GEMMA3 [1B] and the QWEN series perform even weaker under zero-shot settings, with the smallest models (*e.g.*, QWEN2.5-Instruct [0.5B]) falling below 15% F1 on three of the four datasets, indicating that pretraining alone is insufficient for multi-perspective tag generation. In contrast, FLEXTAG effectively bridges this gap through multi-template training and multi-objective optimization.

Tag Selection Performance. Table 2 presents tag selection results on InfiniteBench and HELMET, two OOD test sets. FLEXTAG at 1B scale significantly outperforms GPT-4o on both datasets: FLEXTAG w/ DPO achieves 85.2% EM on InfiniteBench, surpassing GPT-4o by 2.9 percentage points, and 84.6% EM on HELMET, exceeding GPT-4o by 32.0 percentage points. Compared to GPT-4o w/ RAG, FLEXTAG trails by only 6.4% on InfiniteBench, yet surpasses it by 0.9% on HELMET. Given that FLEXTAG has approximately 1% of GPT-4o's parameters, these results demonstrate that targeted training can effectively compensate for the parameter gap on specific tasks. FLEXTAG also exhibits stronger cross-dataset stability: PHI-3.5-vision-instruct [4.2B] achieves 75.0%

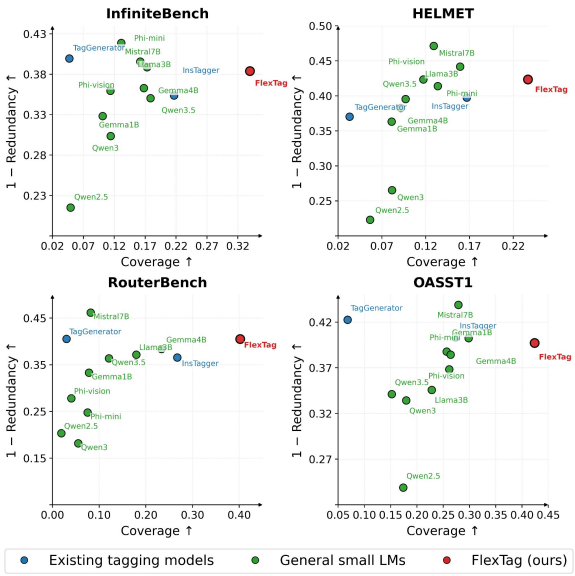

Figure 2: Comparison of multi-perspective performance. X-axis: Coverage (↑); Y-axis: 1-Redundancy (↑). The upper-right region indicates better performance. Colors denote model categories: blue = existing tagging models, green = general small LMs, red = FLEXTAG (ours).

on InfiniteBench but drops sharply to 23.1% on HELMET, whereas FLEXTAG maintains consistently high performance across both datasets. Task-specialized taggers suffer severe generalization failure under template shift, with INSTAGGER [7B] obtaining only 6.9% EM on InfiniteBench and 0% on HELMET, and TAGROUTER [0.5B] failing to produce a single correct prediction on either dataset, directly corroborating the template rigidity problem we identified. Notably, DPO (Rafailov et al., 2023) slightly outperforms GRPO (Shao et al., 2024) on HELMET while the two tie on InfiniteBench, possibly because tag selection is a deterministic choice task where DPO's stable optimization is more suitable.

Multi-perspective Performance. Figure 2 compares model performance along the Coverage and Redundancy dimensions. The X-axis represents Coverage (higher is better), and the Y-axis represents $1 - \text{Redundancy}$ (higher is better). The upper-right region therefore indicates models that both cover diverse tag distributions and avoid semantically redundant predictions. Across all four datasets, FLEXTAG achieves the highest Coverage, with scores of 0.34 on InfiniteBench, 0.24 on HELMET, 0.40 on RouterBench, and 0.42 on OASST1, demonstrating its strong ability to generate broad multi-perspective tag distributions. Existing task-specialized tagging models show limited multi-

Table 3: Ablation study of auxiliary loss terms in FLEXTAG. We report F1 (%) on open-generation tagging. **Bold** denotes the best result.

Ablations	InfiniteBench	HELMET	RouterBench	OASST1
FlexTag [1B]	**51.4**	**43.0**	**58.9**	**59.3**
w/o $\mathcal{L}_{\text{cov.}}$	49.4 ↓2.0	42.5 ↓0.5	57.5 ↓1.4	58.8 ↓0.5
w/o $\mathcal{L}_{\text{syn.}}$	50.6 ↓0.8	39.2 ↓3.8	57.9 ↓1.0	58.6 ↓0.7
w/o $\mathcal{L}_{\text{red.}}$	51.0 ↓0.4	40.6 ↓2.4	58.1 ↓0.8	58.9 ↓0.4

perspective coverage: TAGROUTER (Chen et al., 2025) achieves low Coverage on InfiniteBench and HELMET, indicating that its outputs fail to match the diversity of the ground-truth tag distribution. General-purpose small models also exhibit an unfavorable Coverage–Redundancy trade-off; the QWEN series, for instance, often has low Coverage and high Redundancy, with QWEN3 [0.6B] reaching only 0.18 for $1 - \text{Redundancy}$ on RouterBench. In contrast, FLEXTAG achieves consistently high Coverage while maintaining competitive redundancy, showing the benefit of multi-objective optimization for multi-perspective tagging.

Ablation Studies Table 3 presents the ablation study of the three auxiliary loss terms. Overall, removing any single loss term leads to performance degradation across all four benchmarks, with the effect most pronounced on HELMET. Removing $\mathcal{L}_{\text{syn.}}$ causes the largest single drop, lowering HELMET by 3.8%, and removing $\mathcal{L}_{\text{red.}}$ further drops it by 2.4%, suggesting that synergy and cross-facet redundancy regularization are especially important on this long-context OOD dataset. On InfiniteBench, removing $\mathcal{L}_{\text{cov.}}$ yields the largest degradation (2.0%), whereas removing $\mathcal{L}_{\text{red.}}$ has only a marginal effect (0.4%). The remaining datasets (RouterBench and OASST1) exhibit smaller but consistent drops ($\leq$1.4%). These results suggest that the three regularizers are complementary, and that their benefits are most salient under distribution shift.

Sensitivity Analysis. Figure 3 presents FLEXTAG performance across six backbone models of varying scale. For open generation, most backbones remain usable without catastrophic collapse, although performance varies by backbone: F1 stays within 46.1–58.0% on InfiniteBench and 49.7–66.2% on RouterBench. SMOLLM2 [1.7B] achieves the best F1 on all four datasets, whereas DEEPSEEK-DISTILL [1.5B] (DeepSeek-AI, 2025) is generally the weakest in open generation. For tag selection, the divergence is larger: GEMMA3 [1B] is the most consistent backbone, reaching 85.2% EM on InfiniteBench and 84.6% on HELMET, while QWEN3 [0.6B] drops to 55.7% EM on HELMET. SMOLLM2 [1.7B] (Allal et al., 2025) attains the highest HELMET EM (88.6%) but trails GEMMA3 [1B] on InfiniteBench. Given its balanced performance across open generation and tag selection, we adopt GEMMA3 [1B] as the default backbone.

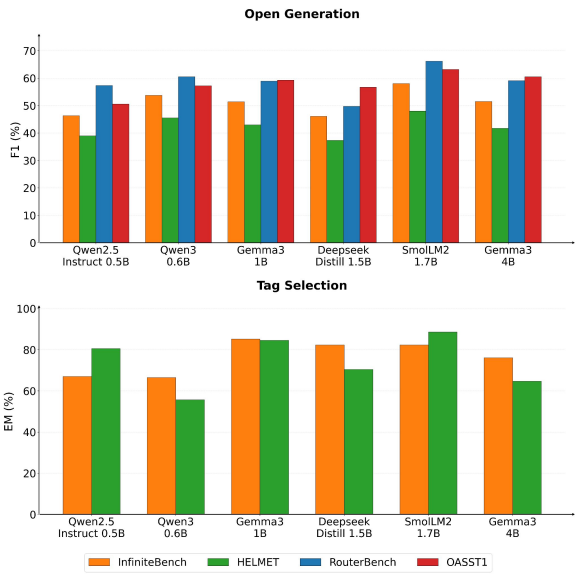

Figure 3: Sensitivity analysis of FLEXTAG across different backbone models. Top: F1 score for open tag generation. Bottom: exact match (EM) for tag selection.

6 Conclusion

We presented FLEXTAG, a 1B-scale instruction tagging framework addressing template rigidity and multi-perspective tag quality in production tagging systems. By combining multi-template augmentation, tag-focused supervision, multi-perspective regularization, and template-adaptive inference, FLEXTAG supports both open-ended tag generation and candidate-constrained tag selection. Under low-resource settings, FLEXTAG outperforms comparable small-model baselines on open generation and surpasses GPT-4o zero-shot under the tag-selection protocol, while approaching the stronger GPT-4o w/ RAG skyline. It also shows robust transfer to source-held-out long-context benchmarks and stable performance on in-domain held-out datasets. Future work will adapt FLEXTAG to specialized domains such as healthcare, law, and finance under scarcer annotation budgets, while preserving tag quality across diverse and evolving schema.

7 Limitations

Our work has several limitations. First, FLEXTAG is designed around a predefined multi-facet tagging schema, which means its effectiveness still depends on the quality and stability of the schema itself. When the target facets are redefined, expanded, or made more hierarchical, additional annotation and adaptation may be required. Second, although FLEXTAG improves robustness to template variation, our experiments mainly study template diversity at the instruction formatting level rather than deeper shifts in task semantics, user intent ambiguity, or domain-specific labeling conventions. Third, the full inference pipeline may introduce extra overhead when constrained decoding and retrieval-augmented few-shot prompting are used together, which could limit deployment in latency-sensitive settings. Finally, while our automatic metrics capture important aspects such as facet coverage and redundancy, they may not fully reflect whether the predicted tags are maximally informative or equally useful for all downstream applications.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, and 3 others. 2025. Smollm2: When smol goes big – data-centric training of a small language model. *Preprint*, arXiv:2502.02737.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and 1 others. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454.

Zhou Chen, Zhiqiang Wei, Yuqi Bai, Xue Xiong, and Jianmin Wu. 2025. Tagrouter: Learning route to llms through tags for open-domain text generation tasks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21539–21564.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, and 1 others. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured nlp tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952.

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

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *Preprint*, arXiv:2305.15717.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *Preprint*, arXiv:2306.11644.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*.

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.

Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, and 1 others. 2023. Openassistant conversations-democratizing large language model alignment. *Advances in neural information processing systems*, 36:47669–47681.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Chen Li, Yixiao Ge, Jiayong Mao, Dian Li, and Ying Shan. 2023. Taggpt: Large language models are zero-shot multimodal taggers. *arXiv preprint arXiv:2304.03022*.

Jianhua Lin. 2002. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. *Preprint*, arXiv:2301.13688.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. # instag: Instruction tagging for analyzing supervised fine-tuning of large language models. *arXiv preprint arXiv:2308.07074*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. *Preprint*, arXiv:2104.08773.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models. In *European Conference on Information Retrieval*, pages 303–313. Springer.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *Preprint*, arXiv:2310.11324.

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*.

Junhong Shen, Neil Tenenholtz, James Brian Hall, David Alvarez-Melis, and Nicolo Fusi. 2024. Tag-llm: Repurposing general-purpose llms for specialized domains. *arXiv preprint arXiv:2402.05140*.

Ruiming Tang, Chenxu Zhu, Bo Chen, Weipeng Zhang, Menghui Zhu, Xinyi Dai, and Huifeng Guo. 2025. Llm4tag: Automatic tagging system for information retrieval via large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 4882–4890.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Praveen Venkatesh, Corbett Bennett, Sam Gale, Tamina Ramirez, Greggory Heller, Severine Durand, Shawn Olsen, and Stefan Mihalas. 2023. Gaussian partial information decomposition: Bias correction and application to high-dimensional data. *Advances in Neural Information Processing Systems*, 36:74602–74635.

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021. Phrase-bert: Improved phrase embeddings from bert with an application to corpus exploration. *Preprint*, arXiv:2109.06304.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. *Preprint*, arXiv:2212.10560.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, and 21 others. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *Preprint*, arXiv:2204.07705.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. *Preprint*, arXiv:2109.01652.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2025. Wizardlm: Empowering large pre-trained language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2024. Helmet: How to evaluate long-context language models effectively and thoroughly. *arXiv preprint arXiv:2410.02694*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and 1 others. 2024. Infinitebench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. Prosa: Assessing and understanding the prompt sensitivity of llms. *Preprint*, arXiv:2410.12405.

A Mathematical Notation

Table 4 summarizes the key mathematical symbols used throughout the paper.

Symbol	Description
Problem Setup	
$x,\ \mathcal{X}$	Input query and its space
$\tau,\ \mathcal{T}$	A template and the set of all templates
$\mathcal{Y}_\tau,\ \mathcal{Y}_\tau^{\text{val}}$	Full output space and valid output subset for template τ
$y = \{(p_j, t_j, e_j)\}_{j=1}^{m}$	Structured output consisting of m perspective-aware tag objects
$p_j,\ t_j,\ e_j$	Perspective, semantic tag, and explanation of the j-th generated item
$\hat{y} = f_\theta(x; \tau)$	Model prediction for input x under template τ
$\mathcal{D}_\tau,\ \mathcal{D}$	Training set for template τ; full dataset $\bigcup_\tau \mathcal{D}_\tau$
f_θ	Parameterized tagger model
Training Objectives	
α	Upweighting factor for tag-value token spans in $\mathcal{L}_{\text{wNLL}}$
$\mathcal{T}_{\text{gen}} \subseteq \mathcal{T}$	Perspective-aware open-generation templates
$\mathcal{P}_\tau$	Set of tracked frequent perspectives under template τ
$\pi_{\tau,p},\ \hat{p}_{\tau,p}$	Empirical and predicted tag distributions for perspective p under template τ
$\mathcal{P}_j(x)$	Token positions assigned to the tag-value span of generated item j
$\hat{p}_j(x)$	Span-level predictive representation of generated item j
$\rho_{j,j'}(x)$	Cosine similarity between predictive representations of two generated items j and j'
$A(y) = \{a_1, \ldots, a_m\}$	Set of perspective–tag items in structured output y
$a_j = (p_j, t_j)$	The j-th perspective–tag item
$s,\ k$	Target and source items sampled for $\mathcal{L}_{\text{syn}}$
$\Delta_{s,k}$	Synergy gap: $\text{sg}(\mathcal{L}_{\text{anchor}}^{(k\to s)}) - \mathcal{L}_{\text{anchor}}^{(\text{all}\to s)}$
$\gamma,\ \lambda$	Margin for $\mathcal{L}_{\text{syn}}$; regularization coefficient for $\mathcal{R}$
$\text{sg}(\cdot)$	Stop-gradient operator
Evaluation & Inference	
σ	Cosine similarity threshold for semantic matching (σ=0.7)
$\mathbf{e}_i$	PhraseBERT embedding of predicted tag t_i
$\hat{p},\ p^*$	Predicted and ground-truth tag distributions for Coverage
$\mathcal{B}_\tau,\ \mathcal{C}_\tau$	Demonstration library and constraint set for template τ

Table 4: Mathematical notation used throughout the paper.

B FlexTag-Bench Pipeline

B.1 Data Sources

FlexTag-Bench is aggregated from 18 publicly available instruction datasets spanning 7 categories, yielding 25,900 raw samples. Table 5 summarizes the data sources and their corresponding sample counts.

Category	Dataset	#Samples
Math	GSM8k	2,000
Code	MBPP	974
	DS1000	1,000
General Conversation	Lima	1,003
	Alpaca	2,000
	Dolly	2,000
	OASST1	2,000
	RouterBench	2,000
	OpenChat	2,000
	Unnatural	2,000
	WizardLM	2,000
PII	pii-masking-200k	2,000
Toxic	lmsys/toxic-chat	2,000
Company Confidential	Contract	423
National Security	ScaleAI/fortress_public	500
Harmful	LLM-LAT/harmful-dataset	2,000

Table 5: Data sources for FlexTag-Bench construction.

B.2 Ground-Truth Annotation

We use GPT-4o to annotate each sample with flexible perspective-aware semantic tags. All annotation prompts share a common base instruction that defines the output schema and the general tagging principle. Specifically, GPT-4o is asked to choose perspectives dynamically from the instruction itself, with at most three high-level perspectives per sample. The perspective set is not fixed, but examples include coding, math, privacy, toxic, security, personal, organization, government, health, finance, education, technology, environment, social, politics, economy, science, engineering, art, music, literature, history, philosophy, religion, and other relevant areas. Multiple tags may be assigned under each perspective, and each tag must include a brief explanation.

On top of this shared base instruction, we append category-specific prompts for different data categories. These category-specific prompts provide additional context about the source data and task format, allowing GPT-4o to adapt its annotations to heterogeneous instruction types while preserving the same perspective-aware JSON schema. All annotations are required to be returned in English and formatted as a valid JSON array of objects: `[{"perspective": str, "tag": str, "explanation": str}]`. After annotation, we further normalize the generated tags through rule-based normalization, frequency filtering, semantic

aggregation, and association-based filtering.

Generate semantic tags from multiple perspectives. Choose the perspectives dynamically from the instruction itself. At most 3 perspectives, and each perspective should be a high-level area of the instruction, such as “coding”, “math”, “privacy”, “toxic”, “security”, “personal”, “organization”, “government”, “health”, “finance”, “education”, “technology”, “environment”, “social”, “politics”, “economy”, “science”, “engineering”, “art”, “music”, “literature”, “history”, “philosophy”, “religion”, “other”. Multiple tags are allowed for each perspective. Explain each tag briefly. Please return in English. Return only valid JSON in this exact format: [{“perspective”: str, “tag”: str, “explanation”: str}]

Table 6: Shared base instruction for GPT-4o perspective-aware tag annotation. Category-specific prompts are appended to this base instruction for different data sources.

For sensitive categories 7 (PII, Toxic, Confidential, Security, Harmful), we add safety constraints to prevent the model from restating sensitive content.

You are a tagging system that labels the key intent of a **PII (Personally Identifiable Information) related** user instruction. {instruction} If the instruction contains **personal-sensitive information**, **do not reveal or restate that information**. Generate semantic tags from multiple perspectives. Choose the perspectives dynamically from the instruction itself. At most 3 perspectives, and each perspective should be a high-level area of the instruction, such as "coding", "math", "privacy", "toxic", "security", "personal", "organization", "government", "health", "finance", "education", "technology", "environment", "social", "politics", "economy", "science", "engineering", "art", "music", "literature", "history", "philosophy", "religion", "other". Multiple tags are allowed for each perspective. Explain each tag briefly. Please return in English. Return only valid JSON in this exact format: ["perspective": str, "tag": str, "explanation": str]"""

Table 7: An example of PII-specific GPT prompt used for ground-truth annotation.

The remaining category-specific templates (Math, Code, Toxic, Confidential, Security, Harmful) follow the same structure with appropriate modifications for their respective content types.

B.3 Quality Filtering

After annotation, we apply quality filtering to remove invalid or low-quality samples. First, we perform **format filtering** by removing samples with JSON parsing failures, empty annotations, or missing required fields in the annotation schema, *i.e.,* `perspective`, `tag`, and `explanation`. Second, we perform **validity filtering** by removing samples whose annotations are empty or cannot be converted into the target training format. After filtering, we retain 22,423 valid instruction–annotation pairs.

C Training Details

To mitigate template rigidity, we rewrite each sample into multiple instruction formats covering four template categories: (1) **Open generation**: Open-ended multi-facet tag generation. (2) **Tag selection**: Selection from a candidate list. (3) **Zero-shot**: Direct inference without examples. (4) **Few-shot**: In-context learning with demonstrations. Each category contains multiple template variants with different instruction phrasings, facet combinations, and output formats (*e.g.*, with or without explanations). In total, we design diverse template variants across the four categories. Below we present one representative example from each category to illustrate the key characteristics.

After augmentation, the dataset contains 46928 samples. We then split the augmented dataset into SFT, RLHF, and test sets, resulting in 28150 SFT samples, 17835 RLHF samples, and 943 test samples. We construct preference optimization data from the RLHF split. For each RLHF sample, we generate candidate outputs using the SFT-trained model and evaluate them against the ground-truth tags. We filter out samples that are too easy or too noisy, removing samples with accuracy above 85% or below 30%. The remaining samples are used to construct preference pairs for DPO and reward-based GRPO training. All training experiments are conducted on NVIDIA L40S GPUs.

Open Generation Template. The model generates complete multi-facet tags without candidate list constraints.

You are a helpful assistant. For the query below, generate all relevant perspective-aware tags. Each output item must include perspective, tag, and explanation. Output must be JSON: {“perspective”: str, “tag”: str, “explanation”: str}. Query: {prompt}

Table 8: Open generation template example.

Tag Selection Template. The model selects the best matching tag from a given candidate list.
Zero-shot Template. Only one example is provided as a format reference, and the model must perform direct inference.

You are a helpful assistant. Please identify the single best perspective-aware tag **from the following list** that matches the user query. Provide an explanation for your choice. Please respond in the JSON format {"perspective": str, "tag": str, "explanation": str}. ## Tag list {candidate_list} Query: {prompt}

Table 9: Tag selection template example.

{examples} Input: {prompt} Output:

Table 10: Zero-shot template example.

Few-shot Template. Multiple examples are provided as in-context learning references.

{task_description} {examples} Now extract: Input: {prompt} Output:

Table 11: Few-shot template example.

By randomly selecting different template types during training, the model learns to perform the same tagging task across multiple instruction formats, thereby maintaining robustness to template variations at inference time.

D Evaluation Protocol

D.1 Ground-Truth Generation and Evaluation Pipeline

For all four evaluation datasets, we follow a unified annotation generation and evaluation pipeline as illustrated in Figure 4.

Ground-Truth Generation. We use GPT-4o as the annotator with a multi-perspective tagging prompt. The prompt instructs the model to generate high-level semantic tags that capture the key heuristics of the input text, emphasizing that tags should be abstract semantic labels useful for categorization or routing decisions, rather than keywords or entities extracted from the text. Raw annotations from GPT-4o are processed through a four-stage normalization pipeline adapted from InsTag: (i) *Frequency Filtering*, which removes tags that appear fewer than 2 times across the dataset; (ii) *Rule-based Aggregation*, which applies lowercase conversion, special character removal, and stemming to standardize tag surface forms; (iii) *Semantic Aggregation*, which computes PhraseBERT embeddings for all tags and then clusters them using DBSCAN with a cosine similarity threshold greater than 0.9 to merge semantically equivalent tags; and (iv) *Association Aggregation*, which merges highly co-occurring tag pairs (co-occurrence > 20, confidence ≥ 0.99) into canonical forms.

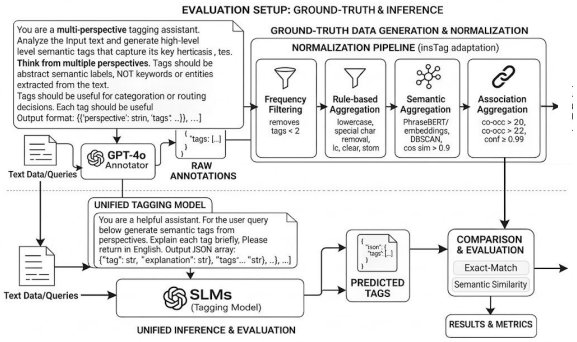

Figure 4: Evaluation pipeline for ground-truth generation, normalization, inference, and comparison.

D.2 Inference Prompts

All models are evaluated using identical prompts to ensure fair comparison. For open generation evaluation, we use a unified prompt that instructs the model to generate multi-perspective semantic tags with explanations (Table 12). For tag selection evaluation, we provide dataset-specific candidate lists: InfiniteBench uses 12 task-oriented tags covering code, QA, math, and retrieval tasks (Table 13), while HELMET uses 7 tags corresponding to its task categories (Table 14).

You are a helpful assistant. For the user query below, generate semantic tags from multiple perspectives. Explain each tag briefly. Please return in English. Output JSON array: [{"perspective": str, "tag": str, "explanation": str}, ...] Query: {truncated_context}

Table 12: Unified prompt template for open generation evaluation.

You are a helpful assistant. Please identify the best tag from the list below that matches the user query. Provide an explanation for your choice. Respond in JSON: {"perspective": str, "tag": str, "explanation": str}. ## Tag list - Tag name: Code Debugging, Description: Asking for which function in code causes an error. - Tag name: Code Programming, Description: Asking for the return value in code. - Tag name: English Multiple Choice, Description: Multiple choice questions using English. - Tag name: English Question Answering, Description: Question answering using English. - Tag name: Chinese Question Answering, Description: Question answering using Chinese. - Tag name: Summarization, Description: Summarization of a text. - Tag name: Character Identification, Description: Identify speakers in dialogue. - Tag name: Math Calculation, Description: Asking for results from mathematical calculations. - Tag name: Math Finding, Description: Find the integer in a list. - Tag name: Number Retrieval, Description: Finding a number from repeated text. - Tag name: PassKey Retrieval, Description: Asking for what is a Pass Key in a text. - Tag name: Key Value Retrieval, Description: Finding the corresponding value from a JSON dictionary. Query: {truncated_context}

Table 13: Tag selection prompt template for InfiniteBench evaluation.

You are a helpful assistant. Please identify the best tag from the list below that matches the user query. Provide an explanation for your choice. Respond in JSON: {"tag": str, "explanation": str}. ## Tag list - Tag name: Synthetic recall, Description: Identify the words that appear most frequently in a long synthetic list. - Tag name: Retrieval augmented generation, Description: Question answering with retrieved documents. - Tag name: Answering with citations, Description: Generate concise answer using only the provided search results. - Tag name: Re-ranking, Description: Rank the document by their relevance to the query. - Tag name: Document Summarization, Description: Novel and legal document summarization. - Tag name: Many-shot in-context learning, Description: Classification with many in-context examples. - Tag name: Long Document Question Answering, Description: Answer the question based on the given story and a question. Query: {truncated_context}

Table 14: Tag selection prompt template for HELMET evaluation.