Model	Summ	Exam	Code	Rewriting	Crea W	Func W	Comm	NLP	Overall
Closed-source Models									
ChatGPT	33.3	40.3	36.6	31.6	48.2	40.4	47.6	45.8	42.7
Claude-2	30.6	36.1	41.7	34.2	48.1	42.5	40.6	48.5	42.4
GPT-4	59.7	51.4	69.2	58.3	66.7	60.4	58.3	65.2	61.9
Open-source Models									
SteamSHP	33.3	29.2	26.7	33.3	40.7	31.3	51.4	51.9	40.6
PandaLM	29.2	33.3	31.7	23.3	43.5	32.9	44.8	48.9	38.9
LLaMA-2-Chat-13B	20.8	27.8	19.2	20.0	31.5	27.5	35.8	31.8	29.0
Vicuna-13B-v1.5	30.6	23.6	35.0	28.3	36.1	37.5	45.5	39.8	37.3
WizardLM-13B-v1.2	22.2	20.8	32.5	19.2	28.7	25.4	29.2	33.0	27.8
LLaMA-2-chat-70B	34.7	33.3	36.7	35.8	51.4	54.2	47.2	47.7	45.9
AUTO-J	**45.8**	**38.9**	**59.2**	**47.5**	**54.6**	**57.1**	**58.0**	**57.6**	**54.8**

Table 1: Agreement rates for pairwise comparison on different scenario groups and overall results. Results with underline are the best among all models and results in **bold** are the second-best. The mapping from abbreviations to names of scenario groups are: Summ → Summarization, Crea W → Creative Writing, Func W → Functional Writing, and Comm → General Communication.

annotated with a reformatted judgment (or discarded), or we have collected 100 samples for this scenario. The final size of pairwise training data is 3,436, and the detailed statistics are in Tab. 21.

Single-response: For single-response, we pick 960 query-response pairs from Chatbot Arena Conversations with a balanced sampling on different scenarios. In preliminary experiments, directly incorporating the scenario criteria as the system message (as in pairwise evaluation) impairs GPT-4's performance on single-response assessment, overly constraining its generated output to the scenario-specific criteria. Therefore, we adopt a "divide-and-conquer" strategy: We collect two pieces of critiques from GPT-4 for a single response with and without scenario criteria as a system message, and then in the third inference, we get the final evaluation judgment by asking GPT-4 to combine these two critiques into a more comprehensive critique and give a final rating. The user message prompt and the prompt for combining critiques are in Tab. 12 and 13, and the detailed statistics are shown in Tab. 22. Tab. 20 shows an example from the "planning" scenario. We find that critiques generated with and without scenario criteria exhibit distinct stylistic differences: The former is longer and closely adheres to the given criteria, whereas the latter is more concise yet capable of incorporating details not covered by the criteria. Finally, combining the above two critiques, a comprehensive critique simultaneously contains general criteria for this scenario and specific details for this sample.

Input format: Besides the collected evaluation judgments, we also need to determine the input format for AUTO-J. In early-stage experiments, we attempted to include the scenario criteria as the system message in the input. However, models trained in this manner performed poorly, often simply paraphrasing the scenario criteria. Therefore, we adopt a technique akin to Context Distillation (Bai et al., 2022b) and Ghost Attention (Touvron et al., 2023b), where we omit the inclusion of scenario criteria in the input for the training data, allowing the model to learn them from the output end implicitly. This design significantly enhances the generality of AUTO-J. The final input formats for pairwise comparison and single-response evaluation are in Tab. 17 and Tab. 18, respectively.

4 TRAINING AUTO-J

By integrating data from both pairwise and single-response evaluations, we train our model to seamlessly toggle between diverse evaluation protocols simply by applying the corresponding prompts. To lessen the positional bias (Wang et al., 2023a) in pairwise comparison, we apply a simple data augmentation trick. For each pairwise training sample, we swap the order of two responses in the input and alternate the "Response 1" and "Response 2" in the evaluation judgment. Since this doubles the pairwise data, we balanced the dataset by duplicating each single-response samples as well.

We train AUTO-J from LLaMA-2-13B-chat (Touvron et al., 2023b) with the DeepSpeed (Rasley et al., 2020) library, Zero Redundancy Optimizer (ZeRO) (Rajbhandari et al., 2020; Ren et al., 2021) Stage 3, gradient-checkpointing (Chen et al., 2016) and FlashAttention (Dao et al., 2022; Dao, 2023) on 8 NVIDIA A100 GPUs. We use the bfloat16 (BF16) and tfloat32 (TF32) mix computation precision