

1	Appendices Contents	
2	A Supplementary Experimental Details	2
3	A.1 Datasets	2
4	A.2 Device & Random Seed	2
5	A.3 Downstream Training	2
6	B Stability Analyses	3
7	B.1 Robustness of Annotation and Criticism Results	3
8	B.2 Impact of Different Sampling Rules on Criticism	3
9	C Examples of Prompt Strategies	4
10	D Theoretical Proofs	7
11	D.1 Proof of Proposition 5.1 - Statistical Properties of ACT Loss (Variance)	7
12	D.2 Proof of Theorem 5.2 - Probabilistic Upper Bound of the Parameter Gap	8
13	E Further Details of ACT Losses	9
14	E.1 ACT Losses with Different Sampling Rules	9
15	E.2 Distributions of Transformed Errors with Different Sampling Rules	9
16	F Sensitivity Analyses of Human Budget	10
17	F.1 AQG vs. Human Budget	10
18	F.2 Downstream Performance vs. Human Budget	11
19	G Related Works and Experimental Comparisons	12
20	G.1 Additional Related Works	12
21	G.2 Potential Improvements Inspired by Related Works	13
22	G.3 Experimental Comparisons	13

23 A Supplementary Experimental Details

24 A.1 Datasets

Table 1: Dataset details.

Dataset	Type	Description	#Classes	Size - Train	Size - Test
CIFAR10	CV	Image classification of basic image categories.	10	50,000	10,000
Fashion-MNIST	CV	Image classification of cloth items.	10	60,000	10,000
Stanford Cars	CV	Image classification of car models.	196	8,143	8,040
TweetEval-Emotion	NLP	Text emotion classification.	4	3,257	1,421
TweetEval-Irony	NLP	Text irony detection.	2	2,862	784
VQA-RAD (Close-end)	VQA	Question-answer pairs on radiology images.	2	940	262

Table 2: Dataset sources and licenses retrieved from <https://paperswithcode.com/datasets>.

Dataset	Source	License
CIFAR10	https://www.cs.toronto.edu/~kriz/cifar.html	N/A
Fashion-MNIST	https://github.com/zalandoresearch/fashion-mnist	MIT
Stanford Cars	https://paperswithcode.com/dataset/stanford-cars	Custom (non-commercial)
TweetEval-Emotion	https://github.com/cardiffnlp/tweeteval	N/A
TweetEval-Irony	https://github.com/cardiffnlp/tweeteval	N/A
VQA-RAD	https://paperswithcode.com/dataset/vqa-rad	CC0 1.0 Universal

25 A.2 Device & Random Seed

26 All experiments are conducted with 1 to 8 NVIDIA SXM5 H100 GPUs with 80GB memories. When
 27 applicable, we set the random seed to 42 for all controllable sources of randomness.

28 A.3 Downstream Training

29 **Sampling** For the thresholding sampling rule, the threshold τ is determined by the quartile corre-
 30 sponding to the human budget proportion. Specifically, we rank the errors in descending order and
 31 set τ as the quartile value that matches the given proportion of the human budget. For the exponential
 32 weighting sampling rule, we try $\beta = 10$ or 100 for all datasets. The mean transition parameter α is
 33 set in the same way as τ , based on the corresponding quartile.

34 **ResNet18** For all datasets, we fine-tune the ResNet18 model initialized with ImageNet-pretrained
 35 weights for 10 epochs. The batch size is set to 4096 for CIFAR-10 and Fashion-MNIST, and 32 for
 36 the Stanford Cars dataset. We use the Adam optimizer for CIFAR-10 and Fashion-MNIST, while
 37 the SGD optimizer is employed for Cars, following the implementation described in [https://](https://www.kaggle.com/code/archanatrivedi/resnet18-on-stanford-car-dataset)
 38 www.kaggle.com/code/archanatrivedi/resnet18-on-stanford-car-dataset. The key
 39 hyperparameters include the learning rate, with a search space of $[1e-2, 1e-3, 5e-4, 1e-4]$, and the
 40 power-tuning parameter for ACT losses, with values selected from $[0.6, 0.7, 0.8, 0.9, 1.0]$ (see
 41 Appendix E for more details about the power-tuning parameter).

42 **RoBERTa** For text classification tasks, we fine-tune the RoBERTa-base model for 5 epochs. We set
 43 the batch size to 32 and use the AdamW optimizer. The key hyperparameters include the learning
 44 rate, with a search space of $[1e-4, 5e-5, 2e-5, 1e-5]$, and the power-tuning parameter for ACT losses,
 45 with values selected from $[0.6, 0.7, 0.8, 0.9, 1.0]$.

46 **BLIP-VQA** For VQA-RAD, we fine-tune the BLIP-VQA model initialized with the blip-vqa-base
 47 for 10 epochs. Since we adopt the close-ended version of VQA-RAD, where answers are limited to
 48 either “Yes” or “No”, we use these two tokens as ground truth. During both training and inference,
 49 the model is prompted to generate either “Yes” or “No” in the response token space. We evaluate
 50 the trained model by checking if the first generated response token—either ‘Yes’ or ‘No’—matches
 51 the correct answer. We use a batch size of 32 and optimize the model using the AdamW optimizer.
 52 The key hyperparameters include the learning rate, searched over $[2e-4, 1e-4, 5e-5, 2e-5]$, and the
 53 power-tuning parameter for ACT losses, selected from $[0.6, 0.7, 0.8, 0.9, 1.0]$.

54 B Stability Analyses

55 B.1 Robustness of Annotation and Criticism Results

56 We assess the robustness of both annotation and criticism by repeating the process 5 times for each
 57 MLLM involved in our explorations. We perform the stability test on a subset of all datasets (i.e., 100
 58 random samples per class). The results are shown in Figure 1, Figure 2, and Figure 3, respectively.
 59 We observe that the standard deviations are generally low (most within 2%). In addition, the rank of
 60 abilities does not change after taking account of potential variations in metric values, indicating that
 61 the conclusions drawn from our explorations are robust to randomness.

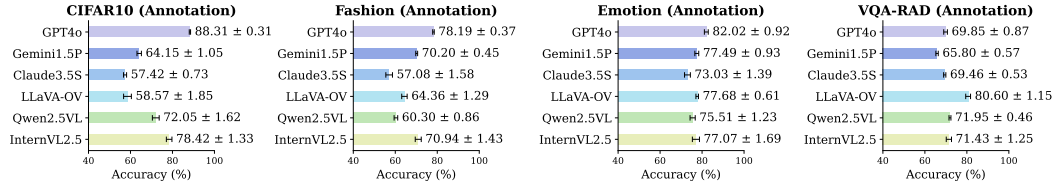


Figure 1: Annotation accuracy with error bars (mean ± std). Results are presented for the CoT prompt strategy across 6 MLLM annotators.

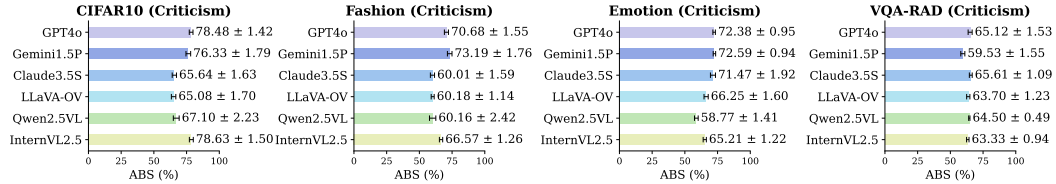


Figure 2: Criticism ABS with error bars (mean ± std). Results are presented for the black-box CoT prompt strategy across 6 MLLM critics.

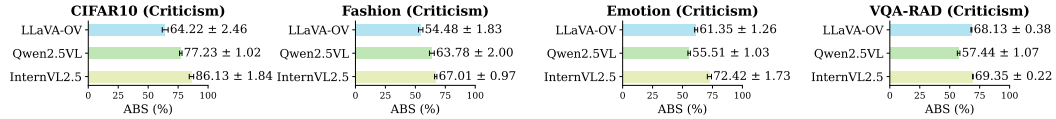


Figure 3: Criticism ABS with error bars (mean ± std). Results are presented for the white-box naïve-logit prompt strategy across 3 MLLM critics.

62 B.2 Impact of Different Sampling Rules on Criticism

63 In Figure 4, we present results computed on the full datasets using different sampling rules to evaluate
 64 the criticism ABS. The results indicate that the choice of sampling rule has minimal impact on the
 65 comparative outcomes. Therefore, the insights derived from our analyses using the thresholding rule
 66 remain consistent across other sampling methods as well.

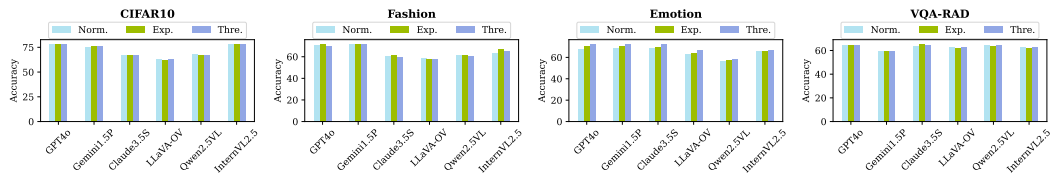


Figure 4: Criticism ABS with various sampling rules. Results are presented for the black-box CoT strategy across 6 MLLM critics.

67 C Examples of Prompt Strategies

68 Here, we present examples of prompt strategies. For annotation prompt - naïve, we include examples
 69 for all three tasks (image classification, text classification, and VQA) to provide a comprehensive
 70 illustration. For other prompt strategies, we only present examples for image classification, but they
 71 can be easily adapted to other tasks by following the same pattern as the annotation prompts. In the
 72 following examples, {purple} denotes inputs, while [blue] denotes outputs. Note that we require
 73 MLLMs to follow a specific output format to simplify the extraction of CoT and labels. In practice,
 74 we observe that MLLMs follow the formatting instructions well.

Illustrations of Inputs and Outputs

- {image_data}, {text_data}, and {question}: Images, texts, and questions from the datasets.
- {label_list_with_index}: A label list with indices of each label, for example, “0: airplane, 1: automobile, 2: bird, 3: cat, 4: deer, 5: dog, 6: frog, 7: horse, 8: ship, 9: truck”.
- {first_label}: The first label in the label list, for example, “airplane”.
- [label_index] & [label_index]: The label index generated by the annotator.
- [error_probability] & [error_level]: The error probability or level generated by the criticizer.
- {CoT_A} & [CoT_A]: The CoT generated by the annotator.
- [CoT]: The CoT generated by the criticizer.

[Image Classification] Annotation Prompt - naïve

Prompt

{image_data}

Determine the label of the image classification task. The list of labels is: {label_list_with_index}. The required output format is: [label_index]. For example, if the label is {first_label}, you should output [0]. Do not return other texts.

Output

[label_index]

[Text Classification] Annotation Prompt - naïve

Prompt

Determine the label of the text classification task. The text is {text_data}. The list of labels is: {label_list_with_index}. The required output format is: [label_index]. For example, if the label is {first_label}, you should output [0]. Do not return other texts.

Output

[label_index]

[VQA-Binary] Annotation Prompt - naïve

Prompt

{image_data}

Answer the question based on the given image. The question is: {question}. The required output format is [0] for No and [1] for Yes. Do not return other text.

Output

[answer_index]

[Image Classification] Annotation Prompt - CoT

Prompt

{image_data}

Determine the label of the image classification task. The list of labels is: {label_list_with_index}. The required output format is: [label_index]. Think step-by-step and provide your reasoning. Example of required output format is: [reasoning][label_index]. The first brackets contain your step-by-step reasoning, and the second brackets contain the label index such as [0] for {first_label}. Do not return other texts.

Output

[CoT_A][label_index]

[Image Classification] Black-box Criticism Prompt - naïve

Prompt

{image_data}

Your task is to produce the probability that the label of the given image is wrong. The list of labels is: {label_list_with_index}. The label of the image is: {label_index}. The required output format is: [error_probability]. For example, [0.911]. The error probability should be reported in 3 decimals. Do not return other texts.

Output

[error_probability]

80

[Image Classification] Black-box Criticism Prompt - CoT

Prompt

{image_data}

Your task is to produce the probability that the label of the given image is wrong. The list of labels is: {label_list_with_index}. The label of the image is: {label_index}. Think step-by-step and provide your reasoning. The required output format is: [reasoning][error_probability]. The first brackets contain your step-by-step reasoning, and the second brackets contain the error probability such as [0.911]. The error probability should be reported in 3 decimals. Do not return other texts.

Output

[CoT][error_probability]

81

[Image Classification] Black-box Criticism Prompt - multiple choice

Prompt

{image_data}

Your task is to analyze if label of the given image is wrong and select from [1: correct, 2: correct but not sure, 3: not sure, 4: incorrect but not sure, 5: incorrect]. The list of labels is: {label_list_with_index}. The label of the image is: {label_index}. Think step-by-step and provide your reasoning. The required output format is: [reasoning][error_level]. The first brackets contain your step-by-step reasoning, and the second brackets contain the error level such as [5] for incorrect. Do not return other texts.

Output

[CoT][error_level]

82

[Image Classification] Black-box Criticism Prompt - devil's advocate

Prompt

{image_data}

Your task is to produce the probability that the statement related to the label of the given image is wrong. The list of labels is: {label_list_with_index}. The statement of the image label is: [CoT_A]. Think step-by-step and provide your reasoning. The required output format is: [reasoning][error_probability]. The first brackets contain your step-by-step reasoning, and the second brackets contain the error probability such as [0.911]. The error probability should be reported in 3 decimals. Do not return other texts.

Output

[CoT][error_probability]

83

84

[Image Classification] White-box Criticism Prompt - naïve

Prompt

{image_data}

Your task is to decide whether the label of the given image is wrong. The list of labels is: {label_list_with_index}. The label of the image is: {label_index}. The required output is either Yes or No, where Yes means mistake and No otherwise. Do not return other texts.

Output

Yes/No

85

[Image Classification] White-box Criticism Prompt - CoT

Prompt

{image_data}

Your task is to decide whether the label of the given image is wrong. The list of labels is: {label_list_with_index}. The label of the image is: {label_index}. Think step-by-step and provide your reasoning. The required output format is: [reasoning][answer]. The first brackets contain your step-by-step reasoning, and the second brackets contain either Yes or No with Yes meaning mistake and No otherwise. Do not return other texts.

Output

[CoT][Yes/No]

86 **D Theoretical Proofs**

87 **D.1 Proof of Proposition 5.1 - Statistical Properties of ACT Loss (Variance)**

88 *Proof.* Recall that the ACT loss is defined as:

$$\mathcal{L}_\theta^{(\text{ACT})} = \frac{1}{N} \sum_{i=1}^N \left(\ell_{\theta,i}^{(m)} + \left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right) \frac{\delta_i(B)}{\pi_B(\hat{\epsilon}_i)} \right), \quad (1)$$

89 where $\delta_i(B) \sim \mathbb{B}(\pi_B(\hat{\epsilon}_i))$. Then, we define

$$Z_i := \ell_{\theta,i}^{(m)} + \left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right) \cdot \frac{\delta_i(B)}{\pi_B(\hat{\epsilon}_i)}, \quad \text{and} \quad Z := \frac{1}{N} \sum_{i=1}^N Z_i = \mathcal{L}_\theta^{(\text{ACT})}.$$

90 Assuming that data are i.i.d, the variance of the ACT loss is now equivalent to the variance of Z ,
91 which is

$$\text{Var}(Z) = \text{Var} \left(\frac{1}{N} \sum_{i=1}^N Z_i \right) = \frac{1}{N^2} \sum_{i=1}^N \text{Var}(Z_i). \quad (2)$$

92 Hence, it suffices to only calculate the variance of Z_i , which can be decomposed into two parts as
93 $\text{Var}(Z_i) = \mathbb{E}[Z_i^2] - (\mathbb{E}[Z_i])^2$. We first expand:

$$\begin{aligned} Z_i^2 &= \left(\ell_{\theta,i}^{(m)} + \left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right) \cdot \frac{\delta_i(B)}{\pi_B(\hat{\epsilon}_i)} \right)^2 \\ &= \left(\ell_{\theta,i}^{(m)} \right)^2 + 2\ell_{\theta,i}^{(m)} \left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right) \cdot \frac{\delta_i(B)}{\pi_B(\hat{\epsilon}_i)} + \left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right)^2 \cdot \frac{\delta_i(B)}{\pi_B(\hat{\epsilon}_i)^2}, \end{aligned} \quad (3)$$

94 where we have $\delta_i^2(B) = \delta_i(B)$ because $\delta_i(B)$ is either 0 or 1. We assume that $\delta_i(B)$ and $\ell_{\theta,i}$ are
95 independent. In addition, it is easy to see that $\mathbb{E}[\delta_i(B)] = \pi_B(\hat{\epsilon}_i)$, and that $\mathbb{E}[Z_i] = \mathbb{E}[\ell_{\theta,i}]$. So, we
96 calculate the expectation of Z_i^2 as follows:

$$\begin{aligned} \mathbb{E}[Z_i^2] &= \left(\ell_{\theta,i}^{(m)} \right)^2 + 2\ell_{\theta,i}^{(m)} \mathbb{E} \left[\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right] + \mathbb{E} \left[\left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right)^2 \cdot \frac{1}{\pi_B(\hat{\epsilon}_i)} \right] \\ &= \mathbb{E} \left[\ell_{\theta,i}^2 - \left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right)^2 \right] + \mathbb{E} \left[\left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right)^2 \cdot \frac{1}{\pi_B(\hat{\epsilon}_i)} \right] \\ &= \mathbb{E} \left[\ell_{\theta,i}^2 \right] + \mathbb{E} \left[\left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right)^2 \cdot \left(\frac{1}{\pi_B(\hat{\epsilon}_i)} - 1 \right) \right] \end{aligned} \quad (4)$$

97 Thus, we have

$$\begin{aligned} \text{Var}(Z_i) &= \mathbb{E} \left[\ell_{\theta,i}^2 \right] + \mathbb{E} \left[\left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right)^2 \cdot \left(\frac{1}{\pi_B(\hat{\epsilon}_i)} - 1 \right) \right] - \mathbb{E}^2[\ell_{\theta,i}] \\ &= \text{Var}(\ell_{\theta,i}) + \mathbb{E} \left[\left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right)^2 \cdot \left(\frac{1}{\pi_B(\hat{\epsilon}_i)} - 1 \right) \right] \end{aligned} \quad (5)$$

98 Finally, we can show that

$$\begin{aligned} \text{Var} \left(\mathcal{L}^{(\text{ACT})} \right) &= \text{Var}(Z) = \frac{1}{N^2} \sum_{i=1}^N \text{Var}(Z_i) \\ &= \frac{1}{N^2} \sum_{i=1}^N \left(\text{Var}(\ell_{\theta,i}) + \mathbb{E} \left[\left(\ell_{\theta,i} - \ell_{\theta,i}^{(m)} \right)^2 \cdot \left(\frac{1}{\pi_B(\hat{\epsilon}_i)} - 1 \right) \right] \right) \\ &= \frac{1}{N} \left(\text{Var}(\ell_\theta) + \mathbb{E} \left[\left(\ell_\theta - \ell_\theta^{(m)} \right)^2 \cdot \left(\frac{1}{\pi_B(\hat{\epsilon})} - 1 \right) \right] \right). \end{aligned} \quad (6)$$

99 This completes the proof. \square

100 D.2 Proof of Theorem 5.2 - Probabilistic Upper Bound of the Parameter Gap

101 *Proof.* From the definition of ACT loss in Equation (1), we can show that

$$\begin{aligned}
\nabla \mathcal{L}_\theta^{(ACT)} &= \frac{1}{N} \sum_{i=1}^N \pi_i \nabla \ell_{\theta,i} + \frac{1}{N} \sum_{i=1}^N (1 - \pi_i) \nabla \ell_{\theta,i}^{(m)} \\
&= \nabla \mathcal{L}_\theta + \left(\frac{1}{N} \sum_{i=1}^N \pi_i \nabla \ell_{\theta,i} - \frac{1}{N} \sum_{i=1}^N \nabla \ell_{\theta,i} \right) + \frac{1}{N} \sum_{i=1}^N (1 - \pi_i) \nabla \ell_{\theta,i}^{(m)} \\
&= \nabla \mathcal{L}_\theta + \frac{1}{N} \sum_{i=1}^N (1 - \pi_i) \left(\nabla \ell_{\theta,i}^{(m)} - \nabla \ell_{\theta,i} \right)
\end{aligned} \tag{7}$$

102 where we let $\pi_i = \frac{\delta_i(B)}{\pi_B(\hat{\epsilon}_i)}$ and $\mathcal{L}_\theta = \frac{1}{N} \ell_{\theta,i}$.

103 It is easy to see $\mathbb{E} \left[(1 - \pi_i) \left(\nabla \ell_{\theta,i}^{(m)} - \nabla \ell_{\theta,i} \right) \right] = 0$. Assume there exist constants $q, C > 0$ such that
104 the transformed error $\pi_B(\hat{\epsilon}_i) \geq q$ for all i with $\delta_i(B) = 1$, and the gradient gap $\left\| \nabla \ell_{\theta,i}^{(m)} - \nabla \ell_{\theta,i} \right\| \leq$
105 C for all $i \in \{1, 2, \dots, N\}$. Next, we bound for any i that

$$\begin{aligned}
\left\| (1 - \pi_i) \left(\nabla \ell_{\theta,i}^{(m)} - \nabla \ell_{\theta,i} \right) \right\| &\leq |1 - \pi_i| \left\| \nabla \ell_{\theta,i}^{(m)} - \nabla \ell_{\theta,i} \right\| \leq \max \left\{ 1, \frac{1-q}{q} \right\} C =: c_0, \\
\text{and } \mathbb{E} \left\| (1 - \pi_i) \left(\nabla \ell_{\theta,i}^{(m)} - \nabla \ell_{\theta,i} \right) \right\|^2 &\leq \mathbb{E} \left[(1 - \pi_i)^2 \right] \left\| \nabla \ell_{\theta,i}^{(m)} - \nabla \ell_{\theta,i} \right\|^2 \leq \frac{1-q}{q} C^2 =: c_1.
\end{aligned}$$

106 Then we apply the vector Bernstein's inequality (e.g., Lemma 18 in [1]) such that

$$\mathbb{P} \left(\left\| \frac{1}{N} \sum_{i=1}^N (1 - \pi_i) \left(\nabla \ell_{\theta,i}^{(m)} - \nabla \ell_{\theta,i} \right) \right\| \geq \epsilon \right) \leq 2 \exp(-N\epsilon^2/(8c_1)) \tag{8}$$

107 for $0 < \epsilon < \frac{c_1}{c_0}$. Then, with a probability of at least $1 - p$ where $p \in (0, 1)$, we have

$$\left\| \frac{1}{N} \sum_{i=1}^N (1 - \pi_i) \left(\nabla \ell_{\theta,i}^{(m)} - \nabla \ell_{\theta,i} \right) \right\| \leq \sqrt{\frac{8c_1 \log(2/p)}{N}} \tag{9}$$

108 for any $N \geq 8c_0^2 \log(2/p)/c_1$.

109 Finally, due to the μ -strong convexity of $\ell_{\theta,i}^{(\cdot)}$ and thus $\mathcal{L}_\theta^{(\cdot)}$, with a probability of at least $1 - p$, we
110 can bound the parameter gap

$$\begin{aligned}
\left\| \theta_*^{(ACT)} - \theta_* \right\| &\leq \frac{1}{\mu} \left\| \nabla \mathcal{L}_{\theta_*^{(ACT)}}^{(ACT)} - \nabla \mathcal{L}_{\theta_*}^{(ACT)} \right\| \\
&= \frac{1}{\mu} \left\| \nabla \mathcal{L}_{\theta_*} - \nabla \mathcal{L}_{\theta_*}^{(ACT)} \right\| \\
&= \frac{1}{\mu} \left\| \frac{1}{N} \sum_{i=1}^N (1 - \pi_i) \left(\nabla \ell_{\theta_*,i}^{(m)} - \nabla \ell_{\theta_*,i} \right) \right\| \\
&\leq \sqrt{\frac{8c_1 \log(2/p)}{\mu^2 N}}
\end{aligned} \tag{10}$$

111 where $\theta_*^{(ACT)} = \arg \min_\theta \mathcal{L}_\theta^{(ACT)}$, and $\theta_* = \arg \min_\theta \mathcal{L}_\theta$.

112 This completes the proof. \square

E Further Details of ACT Losses

E.1 ACT Losses with Different Sampling Rules

The ACT losses with different sampling rules are listed as follows:

- *Normalization* [2, 3]

$$\mathcal{L}_\theta^{(\text{ACT})} = \frac{1}{N} \sum_{i=1}^N \left(\lambda \ell_{\theta,i}^{(m)} + \left(\ell_{\theta,i} - \lambda \ell_{\theta,i}^{(m)} \right) \times \frac{\delta_i(B) \times \sum_{n=1}^N \hat{\epsilon}_n}{B \times \hat{\epsilon}_i} \right); \quad (11)$$

- *Exponential Weighting*

$$\mathcal{L}_\theta^{(\text{ACT})} = \frac{1}{N} \sum_{i=1}^N \left(\lambda \ell_{\theta,i}^{(m)} + \left(\ell_{\theta,i} - \lambda \ell_{\theta,i}^{(m)} \right) \times \delta_i(B) \left(1 + e^{-\beta(\hat{\epsilon}_i - \alpha)} \right) \right); \quad (12)$$

- *Thresholding*

$$\mathcal{L}_\theta^{(\text{ACT})} = \frac{1}{N} \sum_{i=1}^N \left(\lambda \ell_{\theta,i}^{(m)} + \left(\ell_{\theta,i} - \lambda \ell_{\theta,i}^{(m)} \right) \times \delta_i(B) \right), \quad (13)$$

where $\lambda \in [0, 1]$ is the power tuning parameter [2, 4], which controls the extent to which machine annotations are utilized. Specifically, $\lambda = 0$ corresponds to completely ignoring machine annotations, while $\lambda = 1$ corresponds to the full usage. Notably, when employing the thresholding sampling rule with $\lambda = 1$, the ACT loss is equivalent to the standard Cross-entropy loss computed using human-annotated labels when available, and machine-annotated labels otherwise. Therefore, we stated in Section 5.2 that the Cross-entropy loss is a special case of the ACT loss.

E.2 Distributions of Transformed Errors with Different Sampling Rules

In Figure 5, we present the distributions of transformed errors for data samples reviewed by humans ($\delta(B) = 1$) under different sampling strategies. We observe that, with normalization sampling, the lower bounds of the transformed errors are close to 0 across all presented datasets. In contrast, for exponential weighting, the lower bounds typically around 0.8, while thresholding yields a consistent lower bound of 1.0. Based on Theorem 5.2, these results provide an explanation for why exponential weighting and thresholding can lead to better downstream training performance.

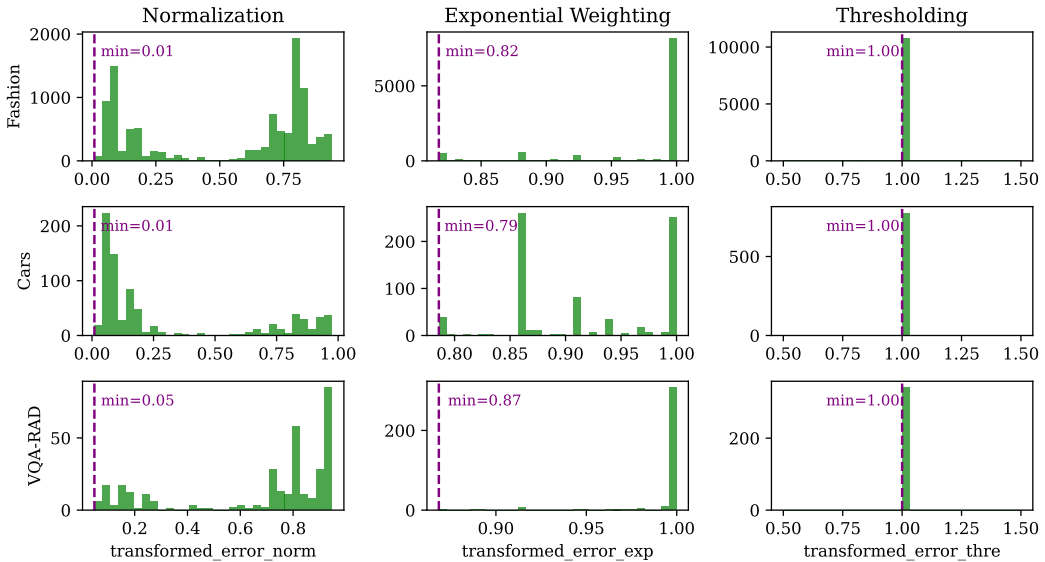


Figure 5: Distributions of transformed errors with different sampling rules ($\delta(B) = 1$). B is set to the ideal human budget for each dataset.

F Sensitivity Analyses of Human Budget

F.1 AQG vs. Human Budget

In Figure 6, we illustrate how Annotation Quality Gain (AQG) changes with the human budget proportion. We observe that as the human budget increases, AQG generally rises rapidly at first, then begins to plateau after a certain point. This initial rapid increase suggests that the human-corrected samples tend to be more obvious and easily identifiable errors. On most datasets, AQG does not reach 100% before the human budget reaches its maximum. In other words, it is usually difficult to achieve perfect annotation quality without reviewing all examples. This indicates that some subtle or hard-to-detect errors are unavoidable. However, we will show in Figure 7 that this does not undermine the effectiveness of using ACT to reduce human effort. With the ACT loss, a promising downstream training does not rely on perfectly labeled data.

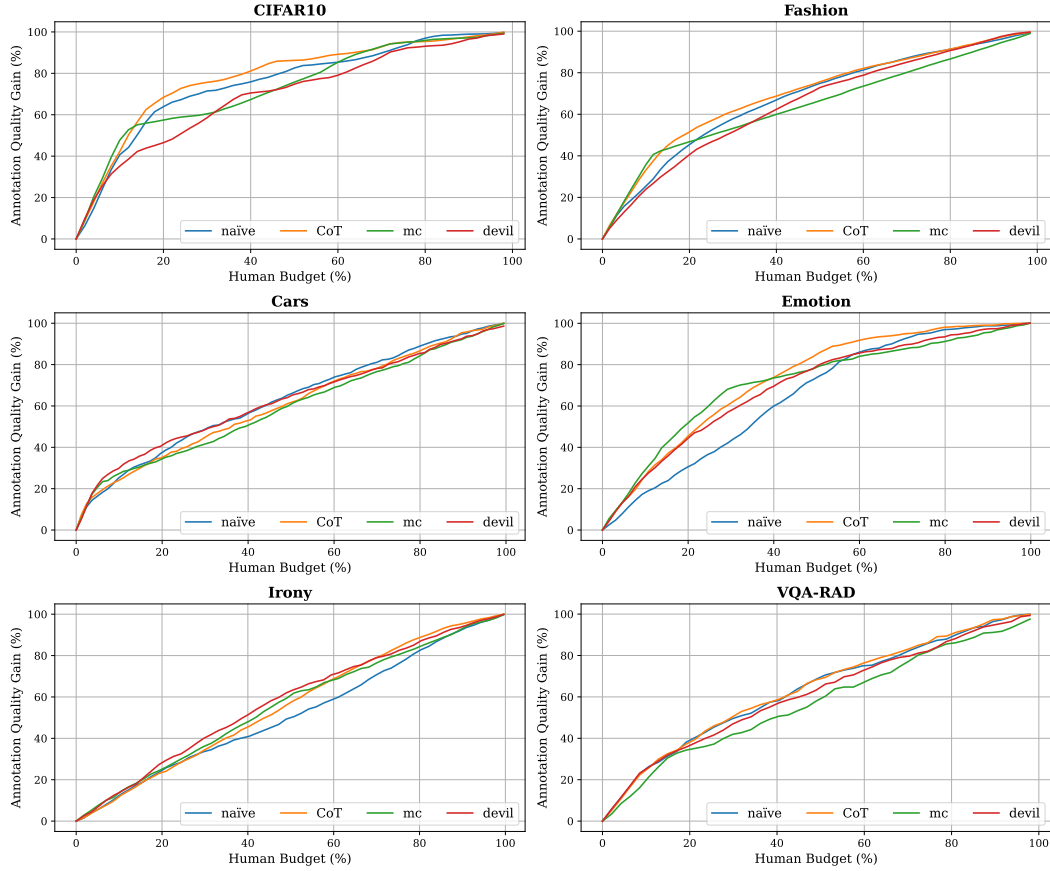


Figure 6: How AQG changes with human budget (%). The results are presented for GPT4o self-criticism with 4 black-box strategies.

F.2 Downstream Performance vs. Human Budget

We conducted the human budget sensitivity experiments using GPT4o self-criticism and the thresholding sampling rule. The results are shown in Figure 7. We observe that both annotation accuracy and downstream accuracy generally increase with a larger human budget. When using the ideal budget, a performance gap can be observed across all datasets. This is because the criticizer is not perfectly accurate, leading to some overlooked mislabeled data, which slightly degrades the final training performance. In Figure 7, we also show the downstream performance gain achieved by adding a 10% buffer budget on top of the ideal budget. In 4 out of 6 datasets, this buffer nearly eliminates the performance gap, while the gap is significantly reduced in the other 2 datasets. **Therefore, we recommend first evaluating the annotator’s accuracy, and then adding a reasonable buffer to the ideal budget based on the observed error rate.**

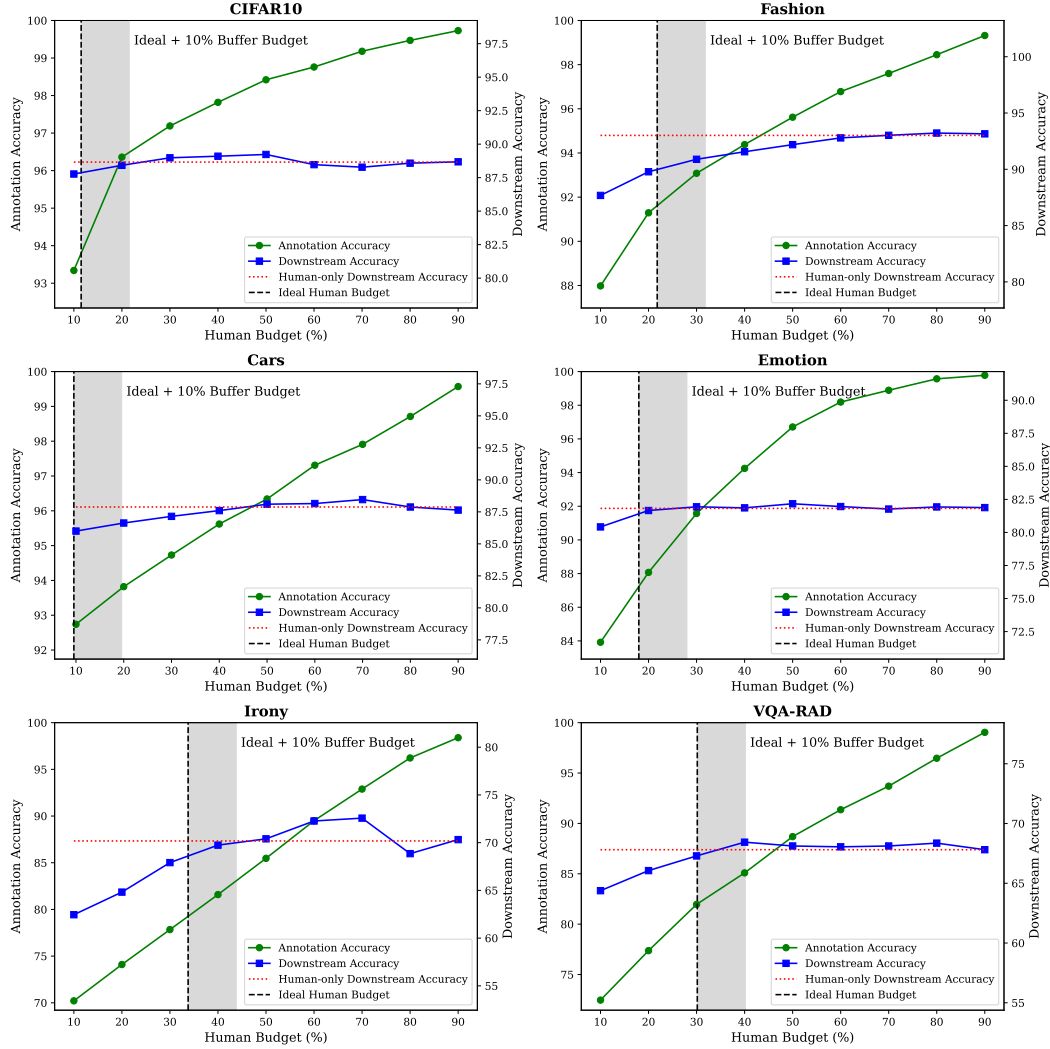


Figure 7: How annotation accuracy and downstream accuracy change with the human budget (%). The black vertical line shows the position of the ideal human budget (i.e., one minus the initial annotator accuracy). The grey area shows the buffer budget over the ideal budget.

G Related Works and Experimental Comparisons

G.1 Additional Related Works

Enhance Data Annotation with LLMs

To improve the quality of LLM annotation, AdaICL adopts in-context learning (ICL) with examples annotated by human [5]. The ICL examples are actively selected based on LLM logit probabilities during annotation, which means their method only supports white-box LLMs. In addition, AdaICL lacks mechanisms for long-context visual inference, making it difficult to apply directly to vision-based tasks. This is because retrieving and encoding a large number of visual examples during inference would incur prohibitive computational costs [6, 7]. However, we consider ICL as a potential direction for future work, particularly as a component in our data pipeline for building an annotator.

Other related works typically follow a three-step LLM-human collaborative framework: (1) LLMs generate initial labels, (2) a verifier assesses the correctness of these labels and outputs verification scores, and (3) human annotators re-annotate a selected subset of labels based on these scores [8, 9, 2, 10]. The primary distinction among these methods lies in the design of the verifier. For instance, Model-in-the-Loop (MILO) [10] utilizes the logit scores from another LLM-based verifier (similar to our white-box criticizer). In contrast, MEGAnno+ [8] directly employs the logit probabilities from the LLM annotator itself. Another framework proposed by [11] uses a verifier implemented as a Support Vector Machine [12], Random Forests [13], or BERT [14], trained on additional human-annotated data. Unlike our approach, the aforementioned methods focus solely on annotation accuracy without considering the utility of annotations for downstream training. This narrow focus limits their effectiveness, as high-accuracy labels do not necessarily translate into meaningful model improvements.

The most relevant work to our research is CDI [2], which identifies LLM errors using a trained XGBoost model [15] and relies on human annotators for correction. During the annotation process, CDI prompts annotators to provide both labels and corresponding verbalized confidence scores. These confidence scores are provided in a black-box manner, where higher values indicate greater confidence. For example, an annotator might respond, “The label is cat, and my confidence is 0.999,” to express high certainty. The XGBoost model then uses these confidence scores as input to learn and predict error probabilities. The ground truth is either 0 or 1 depending on the correctness of the annotation, and then the logit probabilities are regarded as error probabilities. However, CDI has two key limitations: (1) its error detection mechanism lacks flexibility, requiring task-specific design and additional training data, and (2) it employs a normalization-based active M-estimation loss, which we find suboptimal in downstream tasks.

G.2 Potential Improvements Inspired by Related Works

We outline several potential improvements to the ACT data pipeline, inspired by recent related works. First, drawing from AdaICL [5], we could enhance the prompts of the MLLM annotator—particularly for NLP tasks—by incorporating in-context examples. Second, following the approach of MEGAanno+ [8], it may be beneficial to combine the annotator’s confidence scores with the criticizer’s error estimations to better capture the insights from both perspectives. Finally, while the current pipeline relies on a single MLLM annotator–criticizer pair, it could be extended to a multi-model setup using techniques such as majority voting or peer discussion in [16]. An illustration is provided in Figure 8.

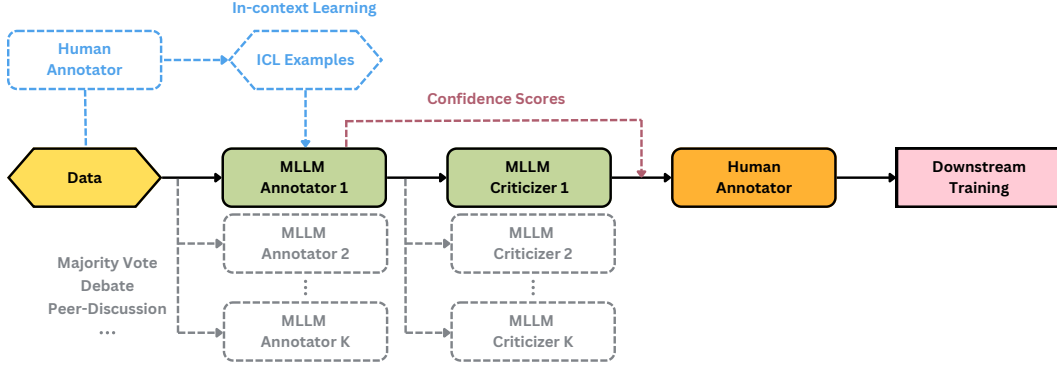


Figure 8: Illustration of potential improvements of the ACT data pipeline.

G.3 Experimental Comparisons

In Table 3, we compare the downstream performance of models trained on ACT and CDI data using various loss functions. Note that CDI data with the ACT norm. loss is corresponding to the approach proposed in [2]. We apply the same human budget (the ideal human budget) to both ACT and CDI. For CDI, a proportion of this budget is allocated to training the XGBoost error detector. The results demonstrate that ACT consistently outperforms CDI in reducing the downstream performance gap.

Table 3: Comparison between ACT and CDI in downstream tasks. The test accuracy (%) is reported in form of mean \pm std over 5 runs.

Training Data - Loss	CIFAR10 (ResNet18)	Fashion (ResNet18)	Cars (ResNet18)	Emotion (RoBERTa)	Irony (RoBERTa)	VQA-RAD (BLIP-VQA)
Human only - Cross-entropy Loss	88.66 \pm 0.97	93.01 \pm 0.63	87.88 \pm 0.36	81.82 \pm 0.57	70.18 \pm 3.23	67.81 \pm 1.47
CDI data - Cross-entropy loss	84.02 \pm 0.85	86.99 \pm 0.72	85.61 \pm 0.24	79.91 \pm 1.37	66.63 \pm 3.44	61.77 \pm 3.41
CDI data - ACT norm. loss	72.22 \pm 1.71	83.38 \pm 1.79	10.76 \pm 1.12	79.05 \pm 1.15	65.96 \pm 3.36	62.24 \pm 3.05
CDI data - ACT exp. loss	84.99 \pm 0.31	87.72 \pm 0.54	86.03 \pm 0.15	80.51 \pm 1.49	68.44 \pm 2.22	67.33 \pm 1.66
CDI data - ACT thre. loss	84.91 \pm 0.57	87.64 \pm 0.52	85.89 \pm 0.27	80.00 \pm 0.83	68.19 \pm 2.29	67.44 \pm 2.16
ACT data - Cross-entropy loss	85.59 \pm 0.52	87.50 \pm 0.86	85.88 \pm 0.26	80.82 \pm 1.08	67.83 \pm 2.82	61.83 \pm 3.27
ACT data - ACT norm. loss	64.70 \pm 5.46	69.27 \pm 7.25	11.54 \pm 0.96	79.87 \pm 0.88	65.66 \pm 2.00	62.55 \pm 3.01
ACT data - ACT exp. loss (Ours)	87.73 \pm 0.36	89.73 \pm 0.35	86.19 \pm 0.14	81.44 \pm 0.51	68.49 \pm 3.20	67.73 \pm 1.33
ACT data - ACT thre. loss (Ours)	87.95 \pm 0.35	89.16 \pm 0.89	86.00 \pm 0.26	81.41 \pm 0.64	68.21 \pm 1.94	67.02 \pm 1.32
Human-CDI performance gap (%)	3.67%	5.29%	1.85%	1.31%	1.74%	0.37%
Human-ACT performance gap (%)	0.71%	3.28%	1.69%	0.38%	1.69%	0.08%
Human budget (%)	11.52%	21.81%	9.56%	17.98%	33.79%	30.15%

References

- [1] Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *International Conference on Machine Learning (ICML)*, pages 1895–1904. PMLR, 2017.
- [2] Kristina Gligorić, Tijana Zrnic, Cino Lee, Emmanuel J Candès, and Dan Jurafsky. Can unconfident llm annotations be used for confident conclusions? *Nations of the Americas Chapter of the Association for Computational Linguistics*, 2025.
- [3] Tijana Zrnic and Emmanuel J. Candès. Active statistical inference. *International Conference on Machine Learning (ICML)*, 2024.
- [4] Anastasios N Angelopoulos, John C Duchi, and Tijana Zrnic. Ppi++: Efficient prediction-powered inference. *arXiv preprint arXiv:2311.01453*, 2023.
- [5] Costas Mavromatis, Balasubramaniam Srinivasan, Zhengyuan Shen, Jiani Zhang, Huzefa Rangwala, Christos Faloutsos, and George Karypis. Which examples to annotate for in-context learning? towards effective and efficient selection. *arXiv preprint arXiv:2310.20046*, 2023.
- [6] Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. LOOK-M: Look-once optimization in kv cache for efficient multimodal long-context inference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4065–4078, 2024.
- [7] Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. Longllava: Scaling multi-modal large language models to 1000 images efficiently via hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2024.
- [8] Hannah Kim, Kushan Mitra, Rafael Li Chen, Sajjadur Rahman, and Dan Zhang. Meganno+: A human-llm collaborative annotation system. *arXiv preprint arXiv:2402.18050*, 2024.
- [9] Xinru Wang, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao. Human-llm collaborative annotation through effective verification of llm labels. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024.
- [10] Yifan Wang, David Stevens, Pranay Shah, Wenwen Jiang, Miao Liu, Xu Chen, Robert Kuo, Na Li, Boying Gong, Daniel Lee, et al. Model-in-the-loop (milo): Accelerating multimodal ai data annotation with llms. *arXiv preprint arXiv:2409.10702*, 2024.
- [11] Yifan Wang, David Stevens, Pranay Shah, Wenwen Jiang, Miao Liu, Xu Chen, Robert Kuo, Na Li, Boying Gong, Daniel Lee, Jiabo Hu, Ning Zhang, and Bob Kamma. Model-in-the-Loop (MILO): Accelerating Multimodal AI Data Annotation with LLMs, September 2024. *arXiv:2409.10702*.
- [12] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [13] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [15] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
- [16] Yu-Min Tseng, Wei-Lin Chen, Chung-Chi Chen, and Hsin-Hsi Chen. Are expert-level language models expert-level annotators? In *NeurIPS 2024 Workshop, Proceedings of Machine Learning Research*, 2024.