Beyond Accuracy Optimization: Computer Vision Losses for Large Language Model Fine-Tuning

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have achieved promising performance on Math Word Problem (MWP) and Question Answering (QA) tasks. LLM fine-tuning is commonly based on cross-entropy loss minimization to perform accurate predictions. However, the standard cross-entropy function neither considers the underlying token distribution over training data nor weighs differently correct and misclassified samples. To address tasks such as closed-ended QA and step-by-step MWP resolution LLMs require advanced language reasoning capabilities. This prompts the adoption of established computer vision loss functions that optimize LLMs' performance rather than simple accuracy. This paper shows the higher effectiveness of combining cross-entropy with computer vision loss functions across MWPs and closed-ended QA datasets. We show relevant LLMs' performance improvements with equal model complexity and the same number of training samples or even fewer. We also demonstrate the efficacy of reproducing step-by-step reasoning on the MWP task.

1 Introduction

Despite their increasing popularity, Large Language Models (LLMs) encounter challenges in handling reading comprehension and formal language understanding tasks such as open-book Question Answering (QA), sentence completion, and Math Word Problems (MWPs). When the task subtends deep reasoning and elaboration of the input question and provides context, LLMs might struggle to achieve competitive performance when used in a zero-shot setting. To specialize deep learning models on specific subtasks, a common practice is to fine-tune the model on in-domain training data. However, this approach typically requires the availability of a large set of human-curated annotations (Min et al., 2024).

Both LLM pre-training and fine-tuning are commonly based on cross-entropy loss minimization. Cross-entropy optimizes the overall system accuracy (Li et al., 2020) but disregards relevant aspects such as the presence of imbalances in token generation and errors. For example, in QA tasks such as MWP a straightforward comparison between the predicted and expected answer fails to capture the text-relevant relations subtended by the formal language (Liu, 2023). When the task involves multiple formal steps, verifying the correctness of the final result does not necessarily guarantee the quality of the LLM outcome. In the worst-case scenario, the reasoning steps are wrong, but the mistakes are concealed by the random selection of the correct answer, yielding an apparently high accuracy score (Turpin et al., 2023).

Inspired by advances in computer vision, in this work, we explore the use of the cross-entropy loss combined with well-known semantic segmentation loss functions in LLM fine-tuning for closed-ended question answering and mathematical reasoning in MWPs. We explore the use of loss functions designed to take into account imbalance in classification (Lin et al., 2017) and optimize performance metrics other than accuracy, such as Generalized Dice Score (Sudre et al., 2017) and Jaccard Index (Berman et al., 2018). The idea behind it is to enhance LLM capabilities by incorporating penalization terms in the loss functions that take token probability distributions and classification error rates into account. The modified objective functions led to consistently better performance without requiring additional training samples or annotations.

The main contributions of the present work are:

- A discussion of the limitations of cross-entropy loss in closed-ended question answering and step-by-step mathematical resolution in MWPs (see Section 3);

- An exploration of the performance of well-known semantic segmentation losses in LLM fine-tuning for five different loss functions, four datasets, and two tasks, even with limited data (see Section 4);

- An extensive performance analysis using MWP reasoning metrics (Golovneva et al., 2022) and a comparison between step-by-step reasoning and accuracy results and an error analysis of common errors done by LLMs in MWP reasoning steps (see Section 4.5).

The source code to reproduce the experiments is available for research purposes at `https://anonymous.4open.science/r/segmentation-losses-nlp-5B73`.

2 Related Works

The main objective of the present work is to explore the use of different loss functions in natural language generation tasks such as closed-ended Question Answering and Math Word Problems. The contribution is rooted in the objective functions that are commonly used for semantic segmentation tasks.

Common loss functions for semantic segmentation. The use of Weighted Cross-Entropy, Dice (Milletari et al., 2016) and Focal losses (Lin et al., 2017) is established for overcoming potential imbalances in the classes to be predicted and to effectively penalize classification errors (Milletari et al., 2016; Berman et al., 2018). The goal is to optimize the overlap between the predicted and ground truth segmentation maps, which can be quantified using the Dice score or the Jaccard Index.

Combining complementary loss functions together (Taghanaki et al., 2019) has shown to improve segmentation performance (Yeung et al., 2022; Shit et al., 2021; Iantsen et al., 2021; Hu et al., 2021c). Transferring a similar approach to text generation tasks is particularly appealing and, to the best of our knowledge, appears to be limited.

Common loss functions for natural language generation. Policy gradient or minimum risk training (Ranzato et al., 2015; Wang et al., 2019) have already been used to optimize the syntactic overlap between generated and expected output, quantified by the BLEU metric (Papineni et al., 2002). Reinforcement Learning suffers from high variance and instability during training. Most efficient solutions rely on soft Q-learning (Guo et al., 2021), differentiable BLEU objectives (Shao et al., 2018, 2021), or EISL loss. The latter is insensitive to the shift of n-grams in target sequences, making it suitable for training with noisy data and weak supervisions; however, its applicability is limited to non-autoregressive models (Liu et al., 2022). In Li et al. (2020), the use of Dice loss and its self-adjusting version has been proposed for the reading comprehension task with encoder-only architectures. However, their benefits depend on the specific task (Li et al., 2020).

Evaluation metrics for reasoning tasks. It is quite common to evaluate the final outcome of the reasoning task regardless of the intermediate steps applied to achieve that result (e.g., (Liang et al., 2022; Cobbe et al., 2021; Hendrycks et al., 2021)). State-of-the-art metrics like ROSCOE (Golovneva et al., 2022) propose new ways to evaluate rationales, although they could not be easily optimized, and the search for representative functions is already opened.

3 Methodology

In this section, we formally introduce the loss functions, shortlisted from the classification presented in Ma et al. (2021), and explain their rationale. For the sake of simplicity, hereinafter we will consider the binary formulation. The loss formulations can be straightforwardly extended to the multi-class scenario.

3.1 Distribution-based losses

This family of loss functions is derived from the Kullback-Leibler Divergence. They aim to optimize the model weights according to the differences between the observed and expected distributions. The traditional cross-entropy loss belongs to this category.

Cross Entropy Loss. Cross-Entropy (CE) is an accuracy-oriented function, i.e., it aims to maximize the accuracy (AC) metric globally in the predicted tokens (Li et al., 2020). AC and CE loss are defined as follows:

$$\text{AC} = \frac{1}{N}\sum_{i}^{N} 1(\hat{y}_i = y_i) \tag{1}$$

$$\text{CE}(p_t) = -\log(p_t) \tag{2}$$

where N is the total number of samples, $\hat{y}_i$ and y_i are the predicted and ground truth class for sample i, respectively, and p_t is the probability of the sample belonging to the positive class.

CE is the most established loss for LLM pre-training and fine-tuning based on Next Token Prediction. Cross-entropy does not consider the underlying structures of predictions or any differences between classes and errors. Imbalance is characteristic of language problems, where the number of classes is equal to the size of a language vocabulary (see Appendix B). Although weighted cross-entropy may address this issue, assigning a proper weight to each class can be challenging.

Focal Loss. Focal loss (FL) is a variant of CE that is specifically designed to address the class imbalance problem. It aims to reduce the relative loss for well-classified examples while emphasizing training on hard, misclassified ones. Focal loss can be defined as follows:

$$\mathrm{FL}(p_t) = -(1-p_t)^{\gamma}\log(p_t) \tag{3}$$

where p_t is the probability of the sample belonging to the positive class while γ is the Focal suppression parameter. Although Focal loss does not directly consider the class distribution, it autonomously distinguishes between hard and easy samples (using $(1-p_t)$). This proves beneficial in correctly predicting underrepresented classes. Notably, this solution gives more importance to errors (i.e., wrongly predicted tokens) than cross-entropy.

3.2 Region-based losses

This family of loss functions optimizes the model's weights according to the differences between two sets.

Dice Loss. It is the main representative of the region-based loss family. The Dice Loss (DL) (Milletari et al., 2016) optimizes the Dice Score (DS) between two sets[1]. They are defined as follows:

$$\mathrm{DS} = \frac{2|\hat{Y}\cap Y|}{|\hat{Y}|+|Y|} = \frac{2TP}{2TP+FP+FN} \tag{4}$$

$$\mathrm{DL} = 1 - \frac{2\sum_i p_i y_i}{\sum_i p_i^2 + \sum_i y_i^2} \tag{5}$$

where $\hat{Y}$ and Y are the prediction and ground truth sets, TP, FP, FN are the numbers of true positives, false positives, and false negatives, respectively, p_i is the probability of the sample belonging to the positive class, and y_i is the ground truth label.

DL directly maximizes a soft version of the Dice Score. It assigns different weights to errors and correct predictions. However, according to Equation (4), correct predictions are deemed more relevant than wrong predictions; therefore, errors may not be sufficiently penalized.

Self-Adjusting Dice Loss. We also evaluate Self-Adjusting Dice Loss (SADL) (Li et al., 2020), which combines the intuitions of Dice and Focal losses. It can be expressed as follows:

$$\mathrm{SADL} = 1 - \frac{2\sum_i (1-p_i)p_i y_i}{\sum_i (1-p_i)p_i + y_i} \tag{6}$$

where the Focal component in Equation (3) is $(1-p_i)$. The rationale behind introducing the Focal component in the Dice Loss is to address the imbalance problem between well-classified and misclassified tokens, which is not adequately covered by Dice Loss.

Generalized Dice Loss. A generalization of the Dice score (Crum et al., 2006) was proposed to consider each class's volume. The corresponding Generalized Dice Loss (GDL) (Sudre et al., 2017) can be expressed as follows:

$$\mathrm{GDL} = 1 - \frac{2\sum_l w_l \sum_i p_{il} y_{il}}{\sum_l w_l \sum_i p_{il} + y_{il}} \tag{7}$$

where $w_l = 1/(\sum_i y_{il})^2$ for each class, while p_i and y_i have the same meanings as defined in Equation (5). This formulation proposes to self-adjust the weight of each class for each sample to address the class imbalance issue.

Lovász Loss. Let $\hat{Y}$ and Y represent the prediction and ground truth sets, respectively. The Jaccard Index (or Intersection-over-Union, IoU) is defined as follows:

$$\mathrm{IoU} = \frac{|\hat{Y}\cap Y|}{|\hat{Y}\cup Y|} = \frac{TP}{TP+FP+FN} \tag{8}$$

Lovász surrogate Loss (LL) has the following expression:

[1] It corresponds to the F1-Score in binary classification.

$$\Delta_{J_1} = 1 - \frac{|\{\hat{Y}=1\} \cap \{Y=1\}|}{|\{\hat{Y}=1\} \cup \{Y=1\}|} \tag{9}$$

$$HL_i(x_i, y_i) = \max(0, 1 - x_i y_i) \tag{10}$$

$$\mathrm{LL} = \overline{\Delta_{J_1}} HL(X, Y) \tag{11}$$

where Δ_{J_1} is the Jaccard loss, HL is the hinge loss, $x_i \in X$ is the prediction logit associated to sample i, $y_i \in Y$ with $y_i \in \{-1, 1\}$, and $\overline{\Delta_{J_1}}$ is the Lovász extension of the Jaccard loss.

LL takes into account both errors and correct predictions. In contrast to Dice loss, which assigns more weight to correctly classified samples, the formulation of Lovász loss allows for an adequate penalty for misclassifications. In many language tasks, we aim not only to penalize errors but also to force the system to avoid introducing extra tokens or omitting certain tokens. This objective can be reached by optimizing the Jaccard Index. We claim that optimizing this objective can be particularly beneficial for the mathematical reasoning task where the model is required to generate the final answer and the intermediate reasoning steps. Specifically, in mathematical reasoning, the intermediate steps must adhere to a stringent structure in terms of syntax (i.e., Math is a formal language) and content (i.e., the sequence of steps required to answer the problem generally lacks many alternative solutions). This makes the task suitable for optimization using Lovász loss.

3.3 Combining loss functions

LLM requires both input and ground truth during training since the training objective is the next token prediction. So, the analyzed tasks require a training set consisting of question (Q) - answer (A) pairs. Let q and a be the number of tokens in Q and A, respectively. We define the fine-tuning language modeling loss as a convex combination (Taghanaki et al., 2019) of CE and one of the different loss functions L under consideration (i.e., FL, GDL, SADL, and LL):

$$\mathcal{L} = \lambda \mathrm{CE}_{Q,A} + (1 - \lambda)\mathrm{L}_A \tag{12}$$

where CE is applied to both the Q's and A's tokens, L is applied only to the A's tokens of the answer, and λ is the mixing parameter (ranging between zero and one).

As depicted in Figure 1, cross-entropy loss is applied to both the question and the answer, as commonly done to ensure good performance in the next token prediction task. This is done following the principle of combo losses, which aims to create more robust training objectives. Conversely, the second component of the loss is applied exclusively to the answer (i.e., ground truth), representing the actual target sequence of interest and following a more rigid structure. Note that the second term of the loss is not applied also to the question tokens as they lack a strict pattern. Consequently, it may incorrectly emphasize underrepresented tokens, which, in this case, are not of interest.

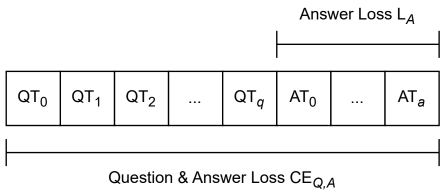

Figure 1: A graphical sketch on how to apply the combined loss on question Q and answer A. *QT*s are question tokens, *AT*s are answer tokens.

3.4 Evaluation Metrics

We consider both standard metrics that consider the final result only (e.g., Exact Match) and metrics that are specifically employed to assess the reasoning steps (suited to MWP only).

Exact Match. Exact Match (EM) is a modified version of the accuracy metric quantifying the similarity between the predicted and expected answers:

$$\mathrm{EM} = \frac{1}{N} \sum_{i}^{N} 1(\hat{A}_i = A_i) \tag{13}$$

where $\hat{A}_i$ is the predicted answer, A_i is the ground truth answer for sample i, and N is the number of samples. Each answer may consist of multiple tokens. The matching is exact if and only if $\hat{A}_i$ and A_i contain precisely the same tokens.

Metrics for the reasoning steps. Our purpose is to check whether the intermediate reasoning steps are correct. To this end, we adopt the ROSCOE metrics (Golovneva et al., 2022) and other metrics, thanks to the systematic and precise nature of mathematical language: Jaccard Index (or IoU, in short) (see Equation (8)); Precision (Prec); Recall (Rec); Dice Score (see Equation (4)); Commutative IoU (C-IoU), which we define as a variant of IoU that accounts for the commutative property of mathematical operations. These metrics are calculated between predicted rationales and ground truth reasoning steps. Unlike ROSCOE, adopting this approach eliminates reliance on external models, thus circumventing potential limitations inherent to the models used.

ROSCOE metrics consider four perspectives: Semantic Alignment, Semantic Similarity, Logical Inference, and Language Coherence. Semantic Alignment (SA) measures the extent to which the generated reasoning is grounded in the source context and aligned with the reference steps, capturing potential hallucinations or missing steps; Semantic Similarity (SS) quantifies the degree of similarity between the generated reasoning and the context or among intermediate steps to identify repetitions; Logical Inference (LI) assesses the internal consistency of the generated reasoning steps and examines for potential contradictions; Language Coherence (LC) evaluates the fluency and grammaticality of the entire reasoning chain. Each metric ranges between zero (worst) and one (best). While, for completeness, we evaluate all the proposed metrics, we argue that LC metrics might not be suitable for assessing mathematical steps, as they are not expressed in natural language.

4 Experimental Results

We perform an extensive experimental evaluation on two tasks for a total of four datasets, five models, and five loss functions. In the following, we summarize the main results. Additional results are available in Appendix D.

4.1 Datasets

We select four datasets, each including at least training and validation sets, therefore neglecting those containing only the test set (being designed for zero-shot benchmarking).

Math World Problems. We consider the following two datasets on MWP: GSM8K (Cobbe et al., 2021) and MathQA (Amini et al., 2019). We have chosen these datasets because they include both the final result and the operational annotations (reasoning steps) leading to the final answer.

GSM8K is included in HELM benchmark (Liang et al., 2022). It collects open-ended questions involving a median of 3 steps to solve them. In contrast, MathQA is designed to operate in a closed-ended QA fashion, with problems involving a median of 4 reasoning steps.

Closed-ended Question Answering. We select two multiple-choice datasets both included in the HELM benchmark, i.e., OpenBookQA (Mihaylov et al., 2018) and HellaSwag (Zellers et al., 2019). We consider these QA datasets because answers are mainly based on reading comprehension rather than relying on prior knowledge of the LLM models.

OpenBookQA comprises questions with multiple choices and contexts to help the reader select the correct answer, whereas HellaSwag provides incomplete sentences with multiple options for appropriate sentence completion.

Detailed information on the considered datasets, including their training/validation/test set splits, are available in Appendix A.

4.2 Models

We employ the following LLMs with a number of parameters ranging from 3B to 7B: RedPajama-Incite-3B (Together Computer, 2023), StableLM-3B (Tow et al.), RedPajama-Incite-7B (Together Computer, 2023), Falcon-7B (Almazrouei et al., 2023), and Llama-2-7B (Touvron et al., 2023). Except for Llama-2 (which is selected as one of the most well-known open-source models), the other ones are selected with the following criteria: (1) They are open-source; (2) They show promising results according to HELM benchmark (Liang et al., 2022); (3) The majority of their training datasets are public or clearly stated to avoid overlapping with analyzed datasets; (4) We consider only the pre-trained version (without any instruction tuning or tuning by human preferences).

More details about the selected models can be found in Appendix C.

4.3 Prompts

We express the prompts to fine-tune the LLM models as follows:
Question: [Question Text] (Context: [Context text])
Answer: [Answer Text]
where *Context* is optional as not every dataset includes it. The answer format can be either a single letter corresponding to the answer for QA or a series of passages and a final answer for mathematical problems. In the latter case, we adhere to the format of GSM8K:
«[Formula]» ... #### [Final answer]
where each *Formula* comprises operators and operands, which can be numbers or symbols. This is done to evaluate better mathematical steps, which exhibit less ambiguity and adhere to stricter lexical

rules than textual reasoning. Prompt examples can be seen in Appendix F.

4.4 Experimental settings

We set the number of training steps to around 25000 and the batch size to 2. We employ the Low Rank Adaptation (Hu et al., 2021a), the AdamW optimizer (Loshchilov and Hutter, 2017), and a linear learning rate scheduler with a warmup of 500 steps. Further information about the experimental settings and implementation details are given in Appendix E.

4.5 Results for the MWP task

Table 1 reports the results achieved on two MWP datasets. We report the macro average of the performance metric achieved by the five models. For the ROSCOE metrics, we report the average for each category (SA, SS, LI, and LC) (Golovneva et al., 2022). Detailed results for each ROSCOE metric are reported in Table 9, whereas additional results are available in Appendix D.

Reasoning step evaluation. On both MWP datasets, the combined loss with Lovász loss (LL) consistently outperforms the Cross-Entropy only setting. It achieves the best performance, likely due the effect of misclassified sample penalties. Specifically, while cross-entropy and Focal loss (FL) aim to maximize global accuracy, LL aims to maximize the global IoU, i.e., it considers both the absence of extra tokens and the presence of missing tokens.

Contrasting results characterize the Self-Adjusting Dice Loss (SADL). It consistently performs better than CE on GSM8K. Conversely, it is less effective than CE on MathQA. This is probably due to the different ways of expressing mathematical operations between the two datasets.

Exact Match. The results obtained on GSM8K show that LLMs adopting the combined loss yield better results than cross-entropy (e.g., LL+CE +1.93% vs. CE).

Correlation analysis between reasoning step metrics. We study the correlation between the ROSCOE metrics and the standard MWP metrics. The goal is to empirically verify whether enhancing LLM reasoning capabilities with ad hoc loss functions has a positive impact on the standard MWP metrics as well. In Table 2 we report the following ROSCOE metrics: Reasoning Alignment (RA), External Hallucination (EH), Redundancy (RD), Common Sense Error (CSE), Missing Step (MS), and Semantic Coverage Chain (SCC). We disregard natural language-oriented metrics, such as the ones related to the language coherence metrics (i.e., grammaticality, perplexity), which are deemed as not relevant to mathematical reasoning.

As expected, all the standard metrics, except for EM (accuracy-oriented), are correlated with the ROSCOE ones, with Pearson correlation values between ≈ 0.5 and ≈ 0.7. This confirms the efficacy of the proposed strategy as jointly optimizing reasoning and final results is beneficial.

General considerations. The results on MathQA and GSM8K show that the final answer tends to be wrong in many cases (low EM values), while the reasoning steps tend to be quite accurate (high or medium-high reasoning step metrics). This highlights that the models generally struggle to correctly predict the final result despite showing a good capability in formulating the mathematical reasoning steps. Often, LLMs identify the exact sequence of reasoning steps needed to solve a mathematical task. However, the predicted final result is wrong since they are not capable of applying the identified steps to compute the final result.

The complete set of results for all metrics and models on the MWP datasets are available in Appendix D, along with statistical tests for significance between cross-entropy and the other loss functions.

Error type analysis in MWP. We also analyze the most common mistakes observed in the MWP reasoning steps. We consider the following metrics covering complementary types of reasoning errors[2]

- Extra Step (ES): proportion of predicted rationales not included in the gold annotations:

$$\text{ES} = \frac{|PS - GTS|}{|PS|} \tag{14}$$

- Missing Step (MS): proportion of gold rationales not generated by the model:

$$\text{MS} = \frac{|GTS - PS|}{|GTS|} \tag{15}$$

- Wrong Operators (WO): proportion of predicted rationales with correct operands but wrong sign according to the gold rationales:

$$\text{WO} = \frac{|PS_{wo}|}{|E|} \tag{16}$$

[2] To the best of our knowledge, there are no standard metrics to evaluate mathematical reasoning.

		Accuracy	Reasoning step Metrics								
		metric	General Purpose metrics					ROSCOE metrics			
	Loss	**EM**	**IoU**	**Prec**	**Rec**	**DS**	**C-IoU**	**SA**	**SS**	**LI**	**LC**
GSM8K	CE	15.83	15.52	19.65	21.43	19.98	19.27	81.14	65.75	34.91	37.58
	FL	15.41 (-0.42)	15.09 (-0.43)	19.27 (-0.38)	21.29 (-0.14)	19.61 (-0.37)	18.71 (-0.57)	81.39 (+0.25)	**66.67** (+0.92)	**36.74** (+1.83)	**37.60** (+0.03)
	GDL	15.00 (-0.83)	15.15 (-0.38)	19.22 (-0.43)	21.23 (-0.20)	19.63 (-0.35)	18.70 (-0.58)	81.08 (-0.06)	65.73 (-0.03)	34.70 (-0.21)	**37.60** (+0.02)
	LL	**17.76** (+1.93)	**17.39** (+1.86)	**21.73** (+2.08)	**23.78** (+2.35)	**22.09** (+2.11)	**21.10** (+1.83)	**81.38** (+0.24)	66.33 (+0.57)	36.00 (+1.09)	37.46 (-0.12)
	SADL	15.91 (+0.08)	15.64 (+0.11)	19.78 (+0.13)	22.35 (+0.91)	20.32 (+0.35)	19.51 (+0.24)	81.33 (+0.18)	66.29 (+0.54)	35.47 (+0.56)	37.62 (+0.05)
MathQA	CE	5.12	36.72	40.30	42.98	40.66	36.78	85.12	68.43	24.21	38.86
	FL	**5.52** (+0.41)	33.73 (-2.99)	37.14 (-3.16)	41.74 (-1.24)	37.98 (-2.68)	33.79 (-2.99)	85.29 (+0.17)	68.39 (-0.05)	23.75 (-0.46)	38.80 (-0.06)
	GDL	5.04 (-0.07)	36.30 (-0.42)	39.35 (-0.95)	44.85 (+1.88)	40.60 (-0.06)	36.36 (-0.42)	85.07 (-0.04)	67.05 (-1.39)	21.01 (-3.20)	38.90 (+0.04)
	LL	4.76 (-0.36)	**43.25** (+6.52)	**46.17** (+5.87)	**50.55** (+7.57)	**47.12** (+6.46)	**43.31** (+6.53)	**85.76** (+0.65)	**70.03** (+1.60)	**28.68** (+4.47)	38.75 (-0.11)
	SADL	4.48 (-0.63)	34.18 (-2.55)	37.69 (-2.62)	43.00 (+0.02)	38.64 (-2.02)	34.23 (-2.55)	84.97 (-0.15)	67.05 (-1.39)	20.42 (-3.79)	**38.95** (+0.09)

Table 1: Macro-average achieved on GSM8K and MathQA datasets. Absolute gains/losses w.r.t. CE results are reported in brackets.

	EM	**IoU**	**Prec**	**Rec**	**DS**	**C-IoU**
RA (SA)	0.1615	0.6582	0.6891	0.6076	0.6739	0.6698
EH (SA)	0.1425	0.6058	0.6186	0.5115	0.5919	0.6074
RD (SA)	0.1607	0.6781	0.6911	0.5674	0.6600	0.6828
CSE (SA)	0.1559	0.5583	0.5314	0.5741	0.5596	0.5608
MS (SA)	0.1744	0.6461	0.6138	0.6595	0.6463	0.6523
SCC (SS)	0.1345	0.5403	0.5501	0.5005	0.5484	0.5495

Table 2: Pearson's correlation between reasoning metrics (ROSCOE) and standard ones (EM, IoU, Prec, Rec, DS, C-IoU) over all samples.

- Inverted Operands (IO): proportion of predicted rationales in which the operands have an incorrect position, considering non-commutative operations:

$$\text{IO} = \frac{|PS_{io}|}{|E|} \tag{17}$$

where GTS and PS are, respectively, the ground truth and predicted reasoning steps, PS_{wo} and PS_{io} are predicted steps with a wrong operator and inverted operands, respectively, and E is the set of errors, i.e., the set of predicted reasoning steps that do not correspond to the gold rationales.

The results are summarized in Table 3. Lovász loss yields the lowest percentages of errors across most error types, particularly in reducing the amount of missing steps. The errors related to wrong operators and inverted operands affect only approximately 4-5% of the reasoning steps for all loss functions. Overall, generating fully accurate reasoning chains remains challenging, but losses such as Lovász loss can help mitigate certain types of errors, making it a preferable training loss than cross-entropy.

Loss	**ES** ↓	**MS** ↓	**WO** ↓	**IO** ↓
CE	67.60%	67.78%	4.68%	5.13%
FL	67.85%	68.48%	**4.22%**	**4.66%**
GDL	68.30%	66.95%	4.57%	5.00%
LL	**62.87%**	**62.83%**	4.27%	**4.66%**
SADL	70.40%	67.32%	4.71%	5.21%

Table 3: Mean errors in mathematical reasoning (see the definitions in Section 4.5 - paragraph entitled *Error type analysis*) across models and datasets.

4.6 Results on Question Answering

We analyze the EM results achieved on the OpenBookQA and HellaSwag datasets. Cross-entropy only proves to be a suboptimal choice in both cases. On OpenBookQA, cross-entropy achieves 75.6, while combining CE with Lovász and Focal losses yields +7.2 and +5.28 improvements, respectively. Conversely, combining CE with Self-adjusting Dice and Generalized Dice losses worsens the performance by -8.20 and -0.20, respectively. On HellaSwag, cross-entropy achieves 47.36. Specifically, Lovász, Focal, and Generalized Dice losses yield $+10.72$, $+24.32$, and $+0.03$ improvements, respectively. In contrast, Self-adjusting Dice loss experiences a decrease in performance of -5.53.

The positive contributions of Focal and Lovász losses are likely due to the fact that FL underestimates the loss contributions of well-predicted samples based on class distribution, whereas Lovász penalizes wrong predictions without suppressing well-predicted samples according to their distributional behavior.

For the sake of completenesse, in Appendix D we report the detailed results for every combination of model and loss as well as the results of the statistical tests for significance.

Results on a reduced number of samples We evaluate the effectiveness of the proposed approach on each task and dataset by reducing the number of training samples to 40% and 10%, while also reducing the training duration by the same amount. In Table 4, we present the mean results for MWP datasets by loss. We show that cross-entropy does not generally yield satisfactory results when the amount of data is reduced. Conversely, losses such as Focal and Lovász demonstrate better capability in extracting desired knowledge even from fewer samples. The same trend is observed in QA datasets, where CE achieves 76.15 and 82.46 for 10% and 40%, respectively. Focal yields improvements of $+5.28$ and $+5.25$, proving the most effective, while Lovász shows improvements of $+1.34$ and $+2.43$. Generalized Dice achieves -3.30 and $+0.07$, while Self-adjusting Dice -0.27 and $+0.68$.

	Loss	CE	GDL	FL	LL	SADL
10%	EM	9.67	9.56	10.08	**10.67**	9.61
	IoU	11.50	11.51	11.94	**12.63**	11.05
	Prec	15.20	15.25	15.62	**16.52**	14.61
	Rec	16.43	16.43	17.04	**17.79**	15.89
	C-IoU	13.16	13.18	13.55	**14.39**	12.75
	DS	15.29	15.30	15.70	**16.56**	14.71
40%	EM	15.48	15.16	17.78	**19.94**	13.62
	IoU	23.60	23.70	24.09	**27.39**	22.48
	Prec	31.52	32.11	32.88	**35.60**	30.49
	Rec	33.31	33.77	34.55	**36.83**	32.26
	C-IoU	29.44	29.93	30.89	**33.18**	28.46
	DS	31.83	32.30	33.07	**35.63**	30.76

Table 4: Results with subsets of training dataset (10% and 40%) on MWP datasets. Results are averaged across models.

5 Conclusion and Future Work

In this work, we explored the application of losses from the semantic segmentation literature to improve Large Language Model efficient fine-tuning for mathematical reasoning and closed-ended question-answering tasks. Our experiments, performed using multiple models across four different datasets, demonstrate that combining cross-entropy with established computer vision losses yields significant performance improvements.

Math Word Problems the LLM fine-tuned with a combination of cross-entropy and Lovász loss achieved the best performance on most reasoning metrics on math word problems, outperforming cross-entropy by over 5-10% absolute in some cases. Standard metrics have shown to be correlated with the state-of-the-art ROSCOE reasoning evaluators. On question-answering datasets such as OpenBookQA and HellaSwag, Lovász and Focal losses consistently outperform cross-entropy. The error analysis revealed that models still struggle with fully accurate reasoning, often missing necessary steps or adding extraneous ones. However, the alternative losses help mitigate certain errors, with Lovász loss yielding the lowest rates of missing and useless steps. Overall, our results illustrate the importance of choosing appropriate loss functions during fine-tuning to optimize end evaluation metrics more effectively. Employing losses tailored to the task of interest can boost performance even without additional data.

Future work can explore the design of new adhoc functions to optimize these tasks and other natural language generation tasks. Moreover, transfer learning to other non-English languages could emphasize the imbalance in token distributions of the target language. Therefore, additional experiments on this stream of research could further support the findings of our study.

Limitations

We analyzed only English language datasets from the mathematical reasoning and reading comprehension domains. Additional experiments on other languages and tasks would strengthen the generalizability of our findings. It's worth noting that we have limited our analysis to existing loss functions in computer vision, which could be suboptimal choices for the tasks under consideration. We analyzed tasks with strong constraints to verify the effectiveness of analyzed loss functions; however, this approach may pose limitations in datasets with more open-ended solutions lacking well-defined patterns.

Ethics Statement

The datasets employed in this study do not contain, from our understanding, any personal information, but they can contain some harmful or inappropriate content. This claim can be extended to the employed models, which could provide non-factual, biased, harmful, or inappropriate answers. Their usage is subject to the limitations stated in their respective technical reports and licenses. Their answers are not intended to offend or harm anyone. Language models have environmental impacts due to high computing requirements during pre-training and fine-tuning. We have made efforts to be computationally responsible by reusing open-sourced pre-trained models and efficient fine-tuning with LoRA (Hu et al., 2021b) methods. The gains from improved losses help amortize the resource costs over higher utility. Overall, we have made reasonable efforts to ensure the transparency and auditability of our experimental methodology.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. 2018. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4413–4421.

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

W.R. Crum, O. Camara, and D.L.G. Hill. 2006. Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE Transactions on Medical Imaging*, 25(11):1451–1461.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923.

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2022. Roscoe: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Han Guo, Bowen Tan, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. 2021. Text generation with efficient (soft) q-learning. *CoRR*, abs/2106.07704.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021a. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021b. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Xiaoling Hu, Yusu Wang, Li Fuxin, Dimitris Samaras, and Chao Chen. 2021c. Topology-aware segmentation using discrete morse theory. In *International Conference on Learning Representations*.

Andrei Iantsen, Dimitris Visvikis, and Mathieu Hatt. 2021. *Squeeze-and-Excitation Normalization for Automated Delineation of Head and Neck Primary Tumors in Combined PET and CT Images*, page 37–43. Springer International Publishing.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL*.

Gabrielle Kaili-May Liu. 2023. Perspectives on the social impacts of reinforcement learning with human feedback.

Guangyi Liu, Zichao Yang, Tianhua Tao, Xiaodan Liang, Junwei Bao, Zhen Li, Xiaodong He, Shuguang Cui, and Zhiting Hu. 2022. Don't take it literally: An edit-invariant sequence loss for text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2055–2078, Seattle, United States. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Jun Ma, Jianan Chen, Matthew Ng, Rui Huang, Yu Li, Chen Li, Xiaoping Yang, and Anne L Martel. 2021. Loss odyssey in medical image segmentation. *Medical Image Analysis*, 71:102035.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2024. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2):30:1–30:40.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732.

Chenze Shao, Xilin Chen, and Yang Feng. 2018. Greedy search with probabilistic n-gram matching for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4778–4784, Brussels, Belgium. Association for Computational Linguistics.

Chenze Shao, Yang Feng, Jinchao Zhang, Fandong Meng, and Jie Zhou. 2021. Sequence-Level Training for Non-Autoregressive Neural Machine Translation. *Computational Linguistics*, 47(4):891–925.

Suprosanna Shit, Johannes C. Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien P. W. Pluim, Ulrich Bauer, and Bjoern H. Menze. 2021. cldice - a novel topology-preserving loss function for tubular structure segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pages 240–248. Springer.

Saeid Asgari Taghanaki, Yefeng Zheng, S Kevin Zhou, Bogdan Georgescu, Puneet Sharma, Daguang Xu, Dorin Comaniciu, and Ghassan Hamarneh. 2019. Combo loss: Handling input and output imbalance in multi-organ segmentation. *Computerized Medical Imaging and Graphics*, 75:24–33.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Jonathan Tow, Marco Bellagente, Dakota Mahan, and Carlos Riquelme. Stablelm 3b 4e1t.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. *CoRR*, abs/1902.10245.

Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. 2022. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95:102026.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Appendices

In this supplementary material, we provide additional details as follows:

- Appendix A: Dataset Statistics
- Appendix B: Token Distribution
- Appendix C: Model Summary
- Appendix D: Extended Results
- Appendix E: Implementation Details
- Appendix F: Prompt Examples

A Dataset Statistics

- OpenBookQA[3] (Mihaylov et al., 2018) contains questions that require multi-step reasoning, use of additional common and commonsense knowledge, and rich text comprehension. OpenBookQA is a new kind of question-answering dataset modeled after open-book exams for assessing human understanding of a subject. The training set contains 4960 samples, validation 500, and test set 500.

- HellaSwag[4] (Zellers et al., 2019) introduced a task of commonsense natural language inference, which consists in selecting the most appropriate conclusion for a sentence from a set of possibilities. It contains 39900 samples in the train set and 10000 in validation (which is employed as the test set since the real one does not have ground truth). It is released under MIT license.

- GSM8K[5] (Cobbe et al., 2021) is a dataset of 8.5K high-quality linguistically diverse grade school math word problems. The dataset was created to support answering questions on basic mathematical problems requiring multi-step reasoning. It has 7470 samples in the training set and 1320 in the test set. It is released under the MIT license.

- MathQA[6] (Amini et al., 2019) is a large-scale dataset of math word problems enhancing AQuA (Ling et al., 2017) providing fully-specified operational programs for each problem. It is released under Apache-2.0 license. It comprises 29800 samples in train, 4480 in validation, and 2990 in test.

B Token Distribution

We report the distribution of tokens across the datasets, highlighting the strong imbalance in tokens in Figure 2. Before the analysis, we exclude all special tokens (25) from the tokenizer. We plot the density against the token identifier in the log scale to better highlight peaks and differences.

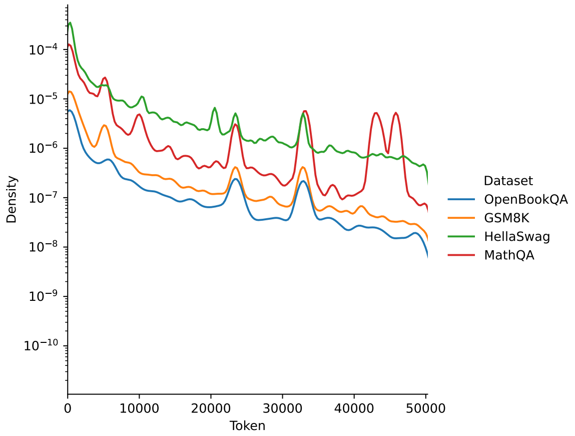

Figure 2: Kernel Density Estimation in log scale for token distributions in GSM8K, MathQA, OpenBookQA, and HellaSwag datasets.

C Model Summary

Table 5 summarizes the characteristics of the models used in this work: RedPajama-Incite-3B[7], StableLM-3B[8], RedPajama-Incite-7B[9], Falcon-7B[10], and Llama-2-7B[11]. For each of them, the following characteristics are reported: model name, number of parameters, license, availability of the pre-training datasets, and mean win rate according to HELM benchmark (Liang et al., 2022).

[3] https://huggingface.co/datasets/openbookqa

[4] https://huggingface.co/datasets/Rowan/hellaswag

[5] https://huggingface.co/datasets/gsm8k

[6] https://huggingface.co/datasets/math_qa

[7] https://huggingface.co/togethercomputer/RedPajama-INCITE-Base-3B-v1

[8] https://huggingface.co/stabilityai/stablelm-3b-4e1t

[9] https://huggingface.co/togethercomputer/RedPajama-INCITE-7B-Base

[10] https://huggingface.co/tiiuae/falcon-7b

[11] https://huggingface.co/meta-llama/Llama-2-7b-hf

Model	# Parameters	License	Pre-Training Datasets	HELM Win Rate
RedPajama-Incite	3B	Apache 2.0	Public	0.311
StableLM	3B	CC BY-SA-4.0	Public	–
RedPajama-Incite	7B	Apache 2.0	Public	0.378
Falcon	7B	Apache 2.0	90% Public	0.378
Llama-2	7B	Llama-2	Public	0.607

Table 5: Model characteristics.

D Extended Results

In the following, we report the extended results for the mathematical reasoning and question-answering tasks.

D.1 Complete results on MWP

In Tables 7 and 8, we present the detailed performance of each model and loss function on MWP datasets. We use McNemar's test for exact match and t-tests (Dietterich, 1998) for other metrics to determine if differences are statistically significant. Using our metrics in GSM8K, Lovász provides the best mean performance across all models, except on Falcon, in which Self-adjusting Dice provide the best ones. Although, they do not show any statistical differences, probably due to the model's limitations. In MathQA, Lovász provides the best performance across most metrics, while regarding the exact match, Focal provides 2 times over 5 the best results. The results for ROSCOE in Table 9 across both MWP datasets show Lovász as the best in most metrics, as highlighted by mean rank, too.

D.2 Complete results on Question Answering

In Table 6, we present the detailed performance of each model and loss function on closed-ended QA datasets. We perform McNemar's test (Dietterich, 1998) to assess whether differences compared to cross-entropy loss are statistically significant. In 9 cases, Lovász loss provides the best improvements in 4 cases, while Focal obtains the best results. The main differences are seen when Lovász fails; Focal still gets improvement. In the inverse case, the results are similar.

E Implementation Details

Based on preliminary experiments, we set the language modeling loss mixing parameter to $\lambda = 0.6$. The Focal suppression parameter was set to $\gamma = 2$. The maximum learning rate was set to $1e-4$ for all datasets, except in GSM8K, for which it is set to $1e-5$.

We selected the checkpoint according to the best validation loss. We train less than 1% of the total parameters using LoRA. During training, the context size is chosen to include most samples without truncation according to 75% percentiles: 128 for GSM8K, MathQA, OpenBookQA, and 256 for HellaSwag. We employ gradient accumulation for context size 256.

We employed Transformers and Peft libraries. Full requirements, versions, and losses' licenses are available in the code repository. For ROSCOE evaluation, we employed the models suggested in the original paper: SimCSE[12] for sentence embedding, RoBERTa[13] for word embedding model, DeBERTa[14] as NLI model, RoBERTa[15] as grammar model, and GPT-2[16] as perplexity model.

We run our experiments on a machine equipped with Intel$^{\circledR}$ Core$^{\text{TM}}$ i9-10980XE CPU, 1 × NVIDIA$^{\circledR}$ RTX A6000 48GB GPU, 128 GB of RAM running Ubuntu 22.04 LTS.

F Prompt Examples

GSM8K *Question: John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. How many hours a week does he spend taking care of dogs? Answer: «10*.5=5» «5*7=35» #### 35*

MathQA *Question: Sophia finished 2 / 3 of a book . she calculated that she finished 90 more pages than she has yet to read . how long*

[12] https://huggingface.co/facebook/roscoe-512-roberta-base

[13] https://huggingface.co/FacebookAI/roberta-base

[14] https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli

[15] https://huggingface.co/cointegrated/roberta-large-cola-krishna2020

[16] https://huggingface.co/openai-community/gpt2-large

Model	**Loss**	**HellaSwag**	**OpenBookQA**
RedPajama 3B	CE	25.26	66.6
	FL	**45.91***	**78.6***
	GDL	25.39	63.8
	LL	26.05	77.2*
	SADL	25.79*	67.0
StableLM 3B	CE	79.69	84.0
	FL	**85.69***	85.4
	GDL	80.0	82.8
	LL	82.97*	**87.2***
	SADL	80.49*	82.4
RedPajama 7B	CE	25.16	74.8
	FL	**73.29***	81.6*
	GDL	25.04	75.8
	LL	25.08	**83.8***
	SADL	25.1	76.6
Falcon 7B	CE	24.59	69.2
	FL	68.51*	77.2*
	GDL	24.94	69.2
	LL	**70.72***	**79.0***
	SADL	26.67*	55.0*
Llama-2 7B	CE	82.12	83.4
	FL	85.03*	81.6
	GDL	81.58	83.8
	LL	**85.6***	**86.8***
	SADL	51.1*	56.0*

Table 6: Results on Question Answering datasets. * indicates values for which $p < 0.05$.

is her book ? Answer: «divide(n0,n1)» «subtract(const_1,#0)» «divide(n2,#1)» #### 270

Model	**Loss**	**EM**	**IoU**	**Prec**	**Rec**	**DS**	**C-IoU**
RedPajama 3B	CE	9.33	11.03	14.66	15.51	14.69	14.76
	FL	9.55	11.46	15.23	16.16	15.33	15.21
	GDL	9.25	11.15	14.81	15.67	14.83	14.92
	LL	**11.45***	**12.52***	**16.66***	**17.17***	**16.52***	**16.53***
	SADL	10.16	11.76	15.80*	16.19	15.60	15.73*
StableLM 3B	CE	24.79	20.96	26.05	26.72	25.93	24.56
	FL	24.79	21.81*	27.36*	27.49	26.95*	25.51*
	GDL	24.87	21.01	26.11*	26.75	25.98	24.58
	LL	**28.66***	**24.02***	**29.42***	**30.38***	**29.38***	**28.15***
	SADL	26.99*	21.08	26.43	27.40	26.39	25.20
RedPajama 7B	CE	**16.07**	15.39	19.93	20.38	19.76	19.76
	FL	14.94	14.93	19.92	19.55	19.32	18.82
	GDL	13.19*	13.94*	18.27*	19.24*	18.33*	17.94*
	LL	16.83	**16.66***	**21.57***	**21.52**	**21.13***	**20.91***
	SADL	13.95*	14.94	19.32	20.41	19.44	18.85
Falcon 7B	CE	4.70	11.39	14.00	20.64	16.15	14.16
	FL	3.49	9.19*	11.25*	19.47	13.69*	11.92*
	GDL	4.40	11.16	13.65	20.85	15.98	13.98
	LL	5.00	11.59	13.93	22.09	16.47	14.08
	SADL	**5.08**	**12.04**	**14.37**	**23.70**	**17.18**	**15.00**
Llama-2 7B	CE	24.28	18.85	23.62	23.92	23.35	23.13
	FL	24.28	18.07	22.61	23.78	22.76	22.07
	GDL	23.29	18.47	23.26	23.64	23.01	22.07
	LL	**26.86***	**22.14***	**27.09***	**27.74***	**26.93***	**25.83***
	SADL	23.37	18.36	22.98	24.03	23.01	22.78

Table 7: Results on GSM8K dataset. * indicates values for which $p < 0.05$.

Model	Loss	EM	IoU	Prec	Rec	DS	C-IoU
RedPajama 3B	CE	**3.47**	30.26	34.20	35.32	34.07	30.29
	FL	2.79	**33.11***	**37.29***	37.87*	**36.88***	**33.16***
	GDL	2.45*	28.98*	32.96*	33.96*	32.72*	29.06*
	LL	2.83	32.83*	36.48*	**38.44***	36.69*	32.86*
	SADL	2.79	26.54*	30.35*	32.55*	30.49*	26.58*
StableLM 3B	CE	8.21	61.98	64.86	67.39	65.36	62.02
	FL	**10.06***	61.98*	65.43*	67.47*	65.66*	62.04*
	GDL	6.86	57.13*	60.16*	63.61*	61.03*	57.16*
	LL	7.50	**65.73***	**68.51***	**70.79***	**69.06***	**65.80***
	SADL	7.16	59.79*	62.85*	65.31*	63.33*	59.84*
RedPajama 7B	CE	7.16	40.35	44.32	45.01	43.98	40.41
	FL	**8.78***	43.12*	47.72*	48.28*	47.16*	43.17*
	GDL	7.05	41.21*	44.87*	45.98*	44.77*	41.27*
	LL	6.82	**46.34***	**49.87***	**51.27***	**49.92***	**46.41***
	SADL	6.10	32.41*	39.17	36.75*	36.79	32.48*
Falcon 7B	CE	5.24	11.34	13.80	21.72	15.93	11.44
	FL	5.84	10.93*	12.98*	24.59*	15.77*	11.00*
	GDL	5.69	11.07*	13.21*	22.98*	15.63*	11.14*
	LL	5.35	**12.77**	**15.00***	**26.07***	**17.67***	**12.87**
	SADL	**5.99**	10.57*	12.62*	21.50*	14.84*	10.63*
Llama-2 7B	CE	1.51	39.69	44.34	45.45	43.98	39.75
	FL	0.15*	19.51*	22.29*	30.48*	24.43*	19.60*
	GDL	**3.17***	43.12*	45.56	57.74*	48.87*	43.16*
	LL	1.28	**58.56***	**61.00***	**66.16***	**62.28***	**58.62***
	SADL	0.38*	41.57*	43.45	58.87*	47.77*	41.62*

Table 8: Results on MathQA dataset. * indicates values for which $p < 0.05$.

	CE	FL	GDL	LL	SADL
Faithfulness	81.96	81.97	81.98	**82.21**	81.96
Informativeness Step	80.61	81.09	**81.11**	80.82	81.10
Faithfulness WW	91.84	92.61	**92.78**	91.55	92.77
Informativeness Chain	90.63	90.40	90.50	**90.79**	90.41
Repetition Word	12.59	13.58	9.80	**15.67**	10.91
Repetition Step	14.44	16.02	12.30	**17.40**	13.30
Reasoning Alignment	92.47	92.37	**92.67**	92.61	92.60
External Hallucination	97.59	97.60	97.57	**97.70**	97.58
Redundancy	88.71	88.60	88.69	**89.06**	88.62
Common Sense Error	97.91	97.87	**97.96**	**97.96**	97.93
Missing Step	89.47	89.47	**89.89**	89.82	89.74
Semantic Coverage Step	98.14	98.25	98.31	**98.32**	98.27
Semantic Coverage Chain	96.21	96.17	**96.36**	96.35	96.30
Discourse Representation	42.71	42.73	41.50	**45.68**	40.95
Perplexity Step	0.28	0.27	**0.28**	0.26	0.27
Coherence Step vs Step	16.41	17.76	14.21	**19.00**	14.94
Perplexity Chain	6.08	6.42	6.74	5.49	**6.84**
Perplexity Step Max	0.14	0.13	0.14	0.14	**0.15**
Grammar Step	94.27	94.18	94.12	**94.28**	94.18
Grammar Step Max	90.32	90.02	89.95	**90.34**	90.00
Mean Rank	3.2	3.45	2.8	**1.95**	3.2

Table 9: Results using ROSCOE metrics aggregated across models and datasets.