Language Generation with Strictly Proper Scoring Rules

Chenze Shao [1] **Fandong Meng** [1] **Yijin Liu** [1] **Jie Zhou** [1]

Abstract

Language generation based on maximum likelihood estimation (MLE) has become the fundamental approach for text generation. Maximum likelihood estimation is typically performed by minimizing the log-likelihood loss, also known as the logarithmic score in statistical decision theory. The logarithmic score is strictly proper in the sense that it encourages honest forecasts, where the expected score is maximized only when the model reports true probabilities. Although many strictly proper scoring rules exist, the logarithmic score is the only local scoring rule among them that depends exclusively on the probability of the observed sample, making it capable of handling the exponentially large sample space of natural text. In this work, we propose a straightforward strategy for adapting scoring rules to language generation, allowing for language modeling with any non-local scoring rules. Leveraging this strategy, we train language generation models using two classic strictly proper scoring rules, the Brier score and the Spherical score, as alternatives to the logarithmic score. Experimental results indicate that simply substituting the loss function, without adjusting other hyperparameters, can yield substantial improvements in model's generation capabilities. Moreover, these improvements can scale up to large language models (LLMs) such as LLaMA-7B and LLaMA-13B. Source code: `https://github.com/shaochenze/ScoringRulesLM`.

1. Introduction

Language generation has played a pivotal role in the advancement of natural language processing, serving as the foundation for a wide range of applications (Bengio et al., 2000; Mikolov et al., 2010; Radford et al., 2018; Brown et al., 2020). The primary goal of language generation is to learn the underlying probability distribution of a given text corpus. To achieve this, maximum likelihood estimation (MLE) is commonly employed to estimate the parameters of a probability distribution that best explains the text corpus (Myung, 2003).

Maximum likelihood estimation is generally performed by minimizing the log-likelihood loss, also known as the logarithmic score, a prominent example of a strictly proper scoring rule (Good, 1952; Gneiting & Raftery, 2007). In statistical decision theory, scoring rules serve as quantitative measures to assess the quality of probabilistic predictions, by assigning a numerical score based on the predicted distribution and the observed sample. A scoring rule is considered strictly proper if it encourages models to report their true beliefs or probabilities. In other words, the expected score is maximized only when the model reports true probabilities, and any deviation from the truth will result in a lower expected score. Due to this property, strictly proper scoring rules are well-suited as loss functions for calibrating probabilistic models (Lakshminarayanan et al., 2017). This is exemplified by the logarithmic score, which corresponds to the log-likelihood loss.

In addition to the logarithmic score, there are other strictly proper scoring rules that provide attractive loss functions for probabilistic prediction problems (Shoemaker, 1991; Hung et al., 1996; Kline & Berardi, 2005; Hui & Belkin, 2021). However, only the logarithmic score has wide applications in language generation, primarily because it is the only strictly proper scoring rule that is also local: it depends exclusively on the predictive probability of the observed sample (Good, 1952; Shuford Jr et al., 1966; Bernardo, 1979). Given the exponentially large sample space for natural text, calculating the score based on the entire probability distribution is infeasible, which hinders the application of non-local scoring rules in language modeling. Consequently, the logarithmic score, being both local and strictly proper, remains the only scoring rule capable of handling the exponentially large sample space of natural text. Nevertheless, the logarithmic score has faced criticism for its unbounded nature and sensitivity to small perturbations in the predicted distribution (Selten, 1998), suggesting that alternative strictly

[1]Pattern Recognition Center, WeChat AI, Tencent Inc. Correspondence to: Chenze Shao <chenzeshao@tencent.com>, Fandong Meng <fandongmeng@tencent.com>, Yijin Liu <yijinliu@tencent.com>, Jie Zhou <withtomzhou@tencent.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

proper scoring rules might offer more suitable and robust options for training and evaluation in specific scenarios.

To investigate the impact and potential benefits of training language models with alternative strictly proper scoring rules, we propose a straightforward strategy for adapting non-local scoring rules to serve as loss functions for language generation. Specifically, we distribute the scoring rule at the token level to promote well-calibrated prediction of conditional probabilities at each time step, consequently leading to well-calibrated sequence-level probability predictions. We further introduce score smoothing to enable honest label smoothing for arbitrary scoring rules. Our approach allows language modeling with any non-local scoring rules while ensuring that the expected loss is minimized only when the model produces the desired probabilities. Leveraging this strategy, we train language generation models using two classic strictly proper scoring rules, the Brier score (Brier, 1950) and the Spherical score (Roby, 1965), as alternatives to the logarithmic score.

Experimental results indicate that simply substituting the loss function, without adjusting other hyperparameters, can yield substantial improvements in the model's generation capabilities. Moreover, these improvements can scale up to large language models (LLMs) such as LLaMA-7B and LLaMA-13B.

2. Strictly Proper Scoring Rules

In this section, we provide essential background on strictly proper scoring rules, including the definition and several popular examples.

2.1. Scoring Rules

Scoring rules assign a numerical score based on the predicted distribution and the observed sample. Let $\mathcal{X} = \{1, ..., m\}$ represents the discrete sample space consisting of a finite number m of different samples, and $\mathcal{P}_m = \{p = (p_1, ..., p_m) : p_1, ..., p_m \geq 0, \sum_{i=1}^m p_i = 1\}$ be the set of probability measures on $\mathcal{X}$. A scoring rule $S(p, i)$ takes values in the extended real line $\overline{\mathbb{R}} = [-\infty, \infty]$, indicating the reward or utility of predicting p when sample i is observed:

$$S(p, i) : \mathcal{P}_m \times \mathcal{X} \mapsto \overline{\mathbb{R}}. \tag{1}$$

Assuming samples conform to a data distribution q, we denote $S(p, q)$ as the expected score:

$$S(p, q) = \mathbb{E}_{i \sim q}[S(p, i)] = \sum_{i=1}^{m} q_i \cdot S(p, i). \tag{2}$$

2.2. Propriety

A scoring rule is proper if the expected score is maximized when the model reports true probabilities:

$$S(p, q) \leq S(q, q), \quad \forall p, q \in \mathcal{P}_m. \tag{3}$$

It is strictly proper when the equality holds if and only if $p = q$. Propriety is an essential requirement for training and evaluating probabilistic models (Bröcker & Smith, 2007; Lakshminarayanan et al., 2017). In terms of training, strictly proper scoring rules can serve as training criteria to calibrate probabilistic models for well-calibrated prediction. In terms of evaluation, strictly proper scoring rules assess the quality of probabilistic predictions by measuring how they align with the true probabilities.

2.3. Locality

A scoring rule is local if the probabilistic prediction is evaluated only at the observed sample, which means that there exists an equivalent function $S_{local}(p_i, i)$ that satisfies:

$$S(p, i) = S_{local}(p_i, i), \quad \forall p \in \mathcal{P}_m, i \in \mathcal{X}. \tag{4}$$

A local scoring rule depends exclusively on the probability of the observed sample, rather than being rewarded for other features of the probabilistic distribution, such as its shape. It has been proven that every scoring rule being both proper and local is equivalent to the logarithmic score (Bernardo, 1979). Formally, if S is both proper and local, then for some constant A and function B, we have:

$$S(p, i) = A \log p_i + B(i). \tag{5}$$

2.4. Examples

We provide some examples of strictly proper scoring rules below.

Logarithmic score. The logarithmic score is a local scoring rule that measures the log probability of the observed sample. It is defined as:

$$S(p, i) = \log p_i. \tag{6}$$

This scoring rule is closely related to maximum likelihood estimation and is widely used in language modeling. Despite its widespread use, the logarithmic score has been criticized for being unbounded and sensitive to small perturbations in the predicted distribution (Selten, 1998).

Brier score. The Brier score (Brier, 1950) is a quadratic scoring rule that measures the mean squared difference between the predicted distribution and the true outcome. It is defined as:

$$S(p, i) = 1 - \sum_{j=1}^{m} (\delta_{ij} - p_j)^2 = 2p_i - \sum_{j=1}^{m} p_j^2, \tag{7}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise. The expected Brier score is $S(p, q) = \sum_{i=1}^{m} q_i^2 - (p_i - q_i)^2$, which is maximized when $p = q$. A more general form is the α-power score (Selten, 1998):

$$S(p, i) = \alpha p_i^{\alpha-1} - (\alpha - 1) \sum_{j=1}^{m} p_j^{\alpha}, \quad \alpha > 1. \tag{8}$$

The α-power score defines a family of strictly proper scoring rules, with the Brier score being a special case for $\alpha = 2$.

Spherical score. The spherical score (Roby, 1965) measures the cosine similarity between the predicted probability vector and the true probability vector. It is defined as:

$$S(p, i) = \frac{p_i}{|p|}. \tag{9}$$

The expected spherical score, $S(p, q) = \langle p, q \rangle / |p|$, is proportional to the cosine similarity and is therefore maximized when $p = q$. A more general form is the pseudo-spherical score:

$$S(p, i) = \frac{p_i^{\alpha-1}}{(\sum_{j=1}^{m} p_j^{\alpha})^{\frac{\alpha-1}{\alpha}}}, \quad \alpha > 1. \tag{10}$$

It reduces to the spherical score when $\alpha = 2$. Note that both the α-power score and the pseudo-spherical score depend on the current prediction probability p_i as well as the global characteristics of the distribution, i.e., the α-norm of p. Therefore, they are strictly proper but non-local.

In addition to the classic scores introduced above, strictly proper scoring rules can also be constructed from any bounded strictly convex function on $\mathcal{P}_m$. Please refer to Gneiting & Raftery (2007) for a literature review.

3. Language Generation with Strictly Proper Scoring Rules

In this section, we present our strategy for adapting non-local scoring rules to serve as loss functions for language generation. Section 3.1 introduces the framework of utilizing scoring rules as loss functions. Section 3.2 describes our approach for distributing the scoring rule at the token level, which overcomes the locality constraint. Section 3.3 further adapts scoring rules to support regularization with label smoothing.

For simplicity of notation, we focus on unconditional sequence models in this section, where samples $x \in \mathcal{X}$ consist of discrete tokens $x = \{x_1, x_2, ..., x_T\}$. The data distribution is represented by $q(x)$, the model predicts the distribution $p_\theta(x)$, and the scoring rule is denoted as $S(p_\theta, x)$. The subsequent discussion can be directly extended to conditional sequence generation scenarios, such as translation and summarization tasks.

3.1. Scoring Rules as Losses

Scoring rules assign a numerical score based on the predicted distribution p_θ and the observed sample x, which can be interpreted as the reward or utility of predicting p_θ when sample x is observed. It is natural to maximize the scoring rule S by minimizing the associated loss function $\mathcal{L}_S$:

$$\mathcal{L}_S(\theta) = -S(p_\theta, q) = -\mathbb{E}_{x \sim q} S(p_\theta, x). \tag{11}$$

As long as S is strictly proper, the associated loss $\mathcal{L}_S$ will have a unique minimizer $p_\theta = q$, encouraging the model to report the true distribution q.

In sequence prediction problems, given the maximum length T_{max} and vocabulary size V, the sample space has an exponentially large size of $V^{T_{max}}$. This makes it intractable to calculate scoring rules that depend on global characteristics of the distribution, such as the Brier score and the spherical score. The logarithmic score, being both local and strictly proper, remains the only scoring rule capable of handling sequence prediction problems. The corresponding loss function is:

$$\mathcal{L}_{log}(\theta) = -\mathbb{E}_{x \sim q} \log p_\theta(x). \tag{12}$$

This loss function can also be derived from maximum likelihood estimation and is commonly referred to as the log-likelihood loss or cross-entropy loss.

3.2. Token-Level Scoring Rules

In general, sequence models do not directly compute the probability of entire sequences. Instead, they decompose the sequence probability into a product of token probabilities in an autoregressive manner:

$$p_\theta(x) = \prod_{t=1}^{T} p_\theta(x_t | x_{<t}). \tag{13}$$

This autoregressive decomposition transforms the sequence prediction task into a series of conditional token prediction tasks, where the sample space is reduced to V for each task. As long as the model predicts the accurate conditional token probability $q(x_t|x_{<t})$, it can correctly recover the sequence probability $q(x)$. Therefore, we can distribute the scoring rule at the token-level to promote well-calibrated prediction for each token prediction task. In this way, we define the following loss based on token-level scoring rules:

$$\begin{aligned} \mathcal{L}_S(\theta) &= -\mathbb{E}_{x \sim q}[\sum_{t=1}^{T} S(p_\theta(\cdot|x_{<t}), x_t)] \\ &= -\sum_{t=1}^{T} \mathbb{E}_{x_{<t} \sim q}[\sum_{x_t} q(x_t|x_{<t}) S(p_\theta(\cdot|x_{<t}), x_t)] \\ &= -\sum_{t=1}^{T} \mathbb{E}_{x_{<t} \sim q}[S(p_\theta(\cdot|x_{<t}), q(\cdot|x_{<t}))]. \end{aligned} \tag{14}$$

In the above equation, $p_\theta(\cdot|x_{<t})$ and $q_\theta(\cdot|x_{<t})$ are probability vectors of size $|V|$, representing the conditional probability distributions of the next word given the history $x_{<t}$. The equation shows that the loss is minimized only when each token-level scoring rule $S(p_\theta(\cdot|x_{<t}), q(\cdot|x_{<t}))$ is maximized. For strictly proper S, maximizing the score means matching every $p_\theta(\cdot|x_{<t})$ with $q(\cdot|x_{<t})$, consequently leading to well-calibrated probability predictions $p_\theta = q$:

$$p_\theta(x) = \prod_{t=1}^{T} p_\theta(x_t|x_{<t}) = \prod_{t=1}^{T} q(x_t|x_{<t}) = q(x). \quad (15)$$

Token-level score optimization allows for language modeling with any non-local strictly proper scoring rules, such as the Brier score (Brier, 1950) and the spherical score (Roby, 1965). For the Brier score, the token-level loss is given by:

$$\mathcal{L}_{Brier}(\theta) = -\mathbb{E}_{x\sim q} \sum_{t=1}^{T} 2p_\theta(x_t|x_{<t}) - |p_\theta(\cdot|x_{<t})|^2. \quad (16)$$

The token-level loss for the spherical score is:

$$\mathcal{L}_{Spherical}(\theta) = -\mathbb{E}_{x\sim q} \sum_{t=1}^{T} \frac{p_\theta(x_t|x_{<t})}{|p_\theta(\cdot|x_{<t})|}. \quad (17)$$

For the logarithmic score, its token-level loss formulation is equivalent to the sequence-level one defined in Equation 12.

3.3. Score Smoothing

In practical applications, it is not always expected for a model to perfectly fit the data distribution, as the label smoothing technique (Szegedy et al., 2016) might be employed for regularization purposes. Label smoothing is typically used in classification and sequence prediction tasks, where it modifies the cross-entropy loss by replacing the one-hot label vector with a soft label to avoid overconfident predictions.

Suppose we have a label set $\mathcal{X} = \{1, ..., m\}$ and a label distribution q. Label smoothing with a smoothing factor ϵ encourages the model to produce a smooth distribution q^ϵ:

$$q_i^\epsilon = (1-\epsilon)q_i + \frac{\epsilon}{m}. \quad (18)$$

The current label smoothing technique is limited to loss functions based on the logarithmic score. Here we introduce a general smoothing technique called score smoothing, which supports label smoothing for arbitrary scoring rules. Given a smoothing factor ϵ, the smoothed score S^ϵ is defined as:

$$S^\epsilon(p, i) = (1-\epsilon) \cdot S(p, i) + \frac{\epsilon}{m} \cdot \sum_{j=1}^{m} S(p, j). \quad (19)$$

Table 1. Expected scores when the model conducts or ignores score smoothing.

Score	Logarithmic	Brier	Spherical
$S(p = q, q^\epsilon)$	$-\infty$	0.8020	0.9010
$S(p = q^\epsilon, q^\epsilon)$	-0.7778	0.8119	0.9011

Ideally, score smoothing should be consistent with the goal of label smoothing, motivating the model to generate the desired smooth distribution q^ϵ. In this context, we define a smoothed score as proper if the expected score satisfies $S^\epsilon(p, q) \leq S^\epsilon(q^\epsilon, q)$, and it is strictly proper when the equality only holds at $p = q^\epsilon$. The following equation shows that S^ϵ is strictly proper as long as S is strictly proper:

$$\begin{aligned} S^\epsilon(p, q) &= (1-\epsilon) \sum_{i=1}^{m} q_i S(p, i) + \frac{\epsilon}{m} \sum_{i=1}^{m} S(p, i) \\ &= \sum_{i=1}^{m} ((1-\epsilon)q_i + \frac{\epsilon}{m}) \cdot S(p, i) \\ &= S(p, q^\epsilon). \end{aligned} \quad (20)$$

By definition, the expected smoothed score $S^\epsilon(p, q) = S(p, q^\epsilon)$ is maximized only when the model produces q^ϵ, proving that S^ϵ is strictly proper as well.

However, when applying score smoothing in practice, we observe that the smoothing term might be ignored in some scoring rules. This is primarily attributed to the corresponding loss being relatively flat around the optimal point q^ϵ. Since a nearly equivalent minimal loss can be achieved without taking the smoothing term into account, the model lacks enough incentive to perform label smoothing.

Consider an example with the number of labels $m = 100$, a one-hot true probability $q = (1, 0, 0, ..., 0)$, and a smoothing factor $\epsilon = 0.1$. Table 1 gives the expected score $S^\epsilon(p, q) = S(p, q^\epsilon)$ when the model produces $p = q, q^\epsilon$ respectively. The logarithmic score imposes $-\infty$ score penalty for disregarding score smoothing with $p = q$. In contrast, the Brier score and the spherical score are bounded, which can only impose a relatively mild penalty when score smoothing is ignored. In particular, the spherical score exhibits nearly identical expected scores in both cases, causing the smoothing term to be almost disregarded.

To address this limitation, we introduce a masked logarithmic score to enhance the smoothing effect. In the target distribution q^ϵ, all labels have a probability of at least $\frac{\epsilon}{m}$. Therefore, labels with probabilities below this threshold can be considered under-smooth. We apply the logarithmic score to further augment the smoothing term for these

Table 2. Implementation details on different datasets.

Dataset	En-De	En-Fr	TED	CNN
batch size	32k	32k	32k	64k
learning rate	7e-4	5e-4	7e-4	2e-4
dropout	0.1	0.1	0.3	0.1
attention dropout	0	0	0	0.1
warmup steps	4k	4k	4k	2k
training steps	200k	300k	18k	100k
fine-tuning steps	50k	50k	4k	20k
weight decay	0	0	0.0	0.01
beam size	5	5	5	4
length penalty	0	0.6	1	2

under-smooth labels:

$$S^{\epsilon}_{\log}(p, i) = S^{\epsilon}(p, i) + \frac{\epsilon}{m} \sum_{j=1}^{m} 1\{p_j < \frac{\epsilon}{m}\} \log p_j, \quad (21)$$

where $1\{\cdot\}$ is the indicator function that takes the value 1 if the inside condition holds. Since the logarithmic score is only applied to under-smooth labels, it does not affect the propriety of the score. Formally, for strictly proper S, we have:

$$S^{\epsilon}_{\log}(p, q) \leq S^{\epsilon}(p, q) \leq S^{\epsilon}(q^{\epsilon}, q) = S^{\epsilon}_{\log}(q^{\epsilon}, q). \quad (22)$$

Therefore, the expected score is maximized only when $p = q^{\epsilon}$, implying that $S^{\epsilon}_{\log}$ is strictly proper. Enhanced by the masked logarithmic score, it ensures a stronger incentive for the model to produce the desired smooth distribution q^{ϵ}.

4. Experiments

4.1. Setup

Datasets. We primarily evaluate our method on machine translation and abstractive summarization. For machine translation, we conduct experiments on widely used translation benchmarks under difference scales: WMT14 English-French (En-Fr, 35.8M pairs), WMT14 English-German (En-De, 4.5M pairs), TED bilingual dataset (10 directions, each with ∼200K pairs). For WMT datasets, we use *newstest2013* for validation and *newstest2014* for test, and apply BPE (Sennrich et al., 2016) with 32K merge operations to learn a joint vocabulary on the tokenized data. For TED bilingual dataset, we use the pre-processed data used in Xu et al. (2021). The translation quality is measured by BLEU (Papineni et al., 2002). For abstractive summarization, We conduct experiments on the summarization benchmark CNN/DailyMail (311K pairs, Hermann et al., 2015). The summarization quality is measured with ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004). We adopt the settings of Transformer-base (Vaswani et al., 2017) for most datasets, except that we use Transformer-big for WMT14 En-Fr. Implementation details are provided in Table 2.

Large Language Models. We further investigate the performance of scoring rules at a larger model scale. Due to the large computational cost of pre-training, we utilize two open-source large language models (LLaMA-7B and LLaMA-13B, Touvron et al., 2023) as our foundation models, and only employ strictly proper scoring rules for instruction tuning. We conduct instruction tuning using the Alpaca dataset by GPT4 (Wang et al., 2022; Taori et al., 2023), which comprises 52K instruction-following demonstrations. We keep the standard settings for instruction tuning on Alpaca, except that the log-likelihood loss is replaced with losses associated with other scoring rules.

Similarly, the generative capability of LLMs is evaluated on machine translation and abstractive summarization. Besides, we also employ MT-bench, a multi-turn question set, to evaluate the open-ended question answering capabilities of LLMs. For machine translation, we follow previous works (Jiao et al., 2023; Zhang et al., 2023a; Zeng et al., 2023; Liu et al., 2023) to evaluate the translation capability on four WMT22 translation tasks (Chinese-to-English, English-to-Chinese, German-to-English, and English-to-German). For text summarization, we follow Liu et al. (2023) to conduct the evaluation on CNN/DailyMail Dataset. We employ beam search with a beam size of 4 for machine translation and 2 for summarization.

4.2. Training from Scratch

In our initial investigation, we evaluate the performance of various strictly proper scoring rules when training language generation models from scratch. We employ three typical scoring rules - the logarithmic score, the Brier score, and the spherical score - to train language generation models. Figure 1 displays their performance curves on three datasets: WMT14 En-De, WMT14 En-Fr, and CNN/DailyMail.

The results indicate that, although all of these scoring rules are strictly proper, they still exhibit noticeable differences when training language generation models from scratch. Among the three datasets, the logarithmic score consistently converges the fastest and achieves the best performance. The spherical score follows, and the Brier score exhibits the slowest convergence and the lowest performance.

We hypothesize that such differences may be attributed to two primary factors. On one hand, despite sharing the same optimum of $p = q$, different strictly proper scoring rules possess distinct learning dynamics. For a specific neural architecture, the optimization trajectory and achievable optimum for each score vary, depending on the characteristics of the score's gradient. For instance, compared to the other two scores, the logarithmic score exhibits a larger gradient during the initial stages of training, which may facilitate model warmup and enable faster convergence. On the other hand, the hyperparameter settings we employed were ob-

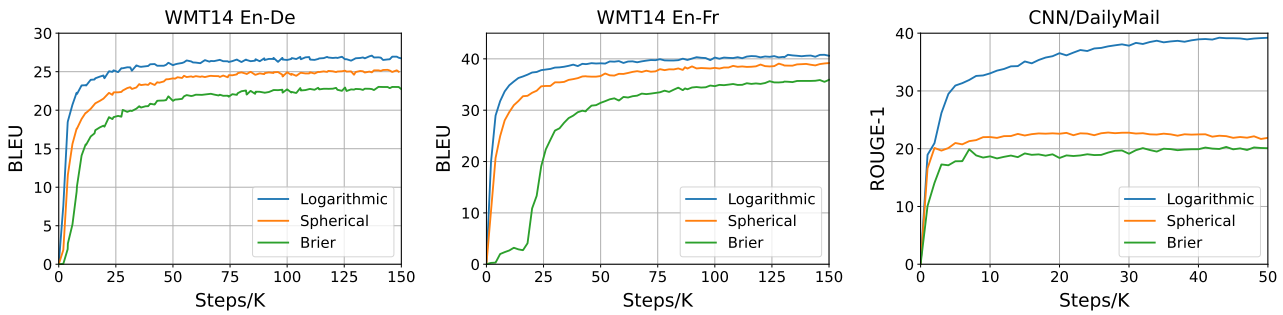

Figure 1. Performance curves of different strictly proper scoring rules on translation and summarization tasks.

Table 3. BLEU scores on WMT14 En-De and WMT14 En-Fr test sets. '+ Brier' and '+ Spherical' represent fine-tuning with the Brier score or the Spherical score. The compared methods are based on our implementation. Statistical significance is indicated by $^*(p < 0.01)$ vs. the baseline.

Model	EN-DE	EN-FR
Transformer	27.61	41.92
MixCE (Zhang et al., 2023b)	27.75	42.03
TaiLr (Ji et al., 2023)	27.95	42.12
Convex (Shao et al., 2023)	27.80	42.05
Transformer + Brier	28.01*	**42.50***
Transformer + Spherical	**28.07***	42.09

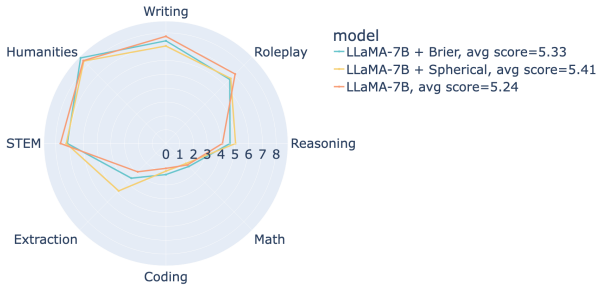

Figure 2. Question answering capabilities evaluated on MT-bench, a multi-turn question set.

tained from previous works that used the logarithmic score for training (Vaswani et al., 2017). These settings may not be as well-suited for other scoring rules, resulting in their relatively inferior performance.

4.3. Fine-tuning with Scoring Rules

As we have already observed, it is relatively challenging for other scoring rules to surpass the performance of the logarithmic score when training from scratch. Here, we further explore the impact of using alternative scores for fine-tuning on models trained with the logarithmic score. We fine-tune from an earlier checkpoint to ensure the total number of training steps remains unchanged. We fix all hyperparameters and only modify the loss function to correspond with the alternative scores.

Table 4. ROUGE scores on CNN/DailyMail. RG-1, RG-2, RG-L stand for ROUGE-1, ROUGE-2, and ROUGE-L scores, respectively. The compared methods are based on our implementation.

Model	RG-1	RG-2	RG-L
Transformer	39.72	17.00	36.41
MixCE (Zhang et al., 2023b)	40.16	17.48	**36.85**
TaiLr (Ji et al., 2023)	39.11	15.99	36.06
Convex (Shao et al., 2023)	40.15	**17.67**	36.70
Transformer + Brier	**40.20**	17.56	36.78
Transformer + Spherical	**40.20**	17.55	36.73

Table 5. BLEU scores of Alpaca fine-tuned large language models on WMT22 test sets.

Model	EN-DE	DE-EN	EN-ZH	ZH-EN
LLaMA-7B	25.42	17.93	13.86	13.17
+ Brier	**29.15**	**21.09**	15.74	17.75
+ Spherical	29.07	21.05	**15.87**	**17.95**
LLaMA-13B	29.35	21.74	15.58	16.27
+ Brier	29.54	22.80	**17.10**	**19.99**
+ Spherical	**29.82**	**23.11**	15.85	19.59

Table 3 and Table 4 present the impact of fine-tuning on translation and summarization performance. As can be seen, even without adjusting hyperparameters for specific scores, fine-tuning with Brier score or Spherical score can still yield certain improvements on logarithmic score pre-trained models. We conjecture that such improvements stem from the complementarity between scoring rules. As different scoring rules follow unique optimization trajectories towards the same global optimum, fine-tuning with another score might aid the model in escaping its current trapped region, consequently leading to further performance improvements.

We continue to explore the effectiveness of scoring rules on a larger model scale. During the instruction tuning of LLaMA-7B and LLaMA-13B, we substitute the log-likelihood loss

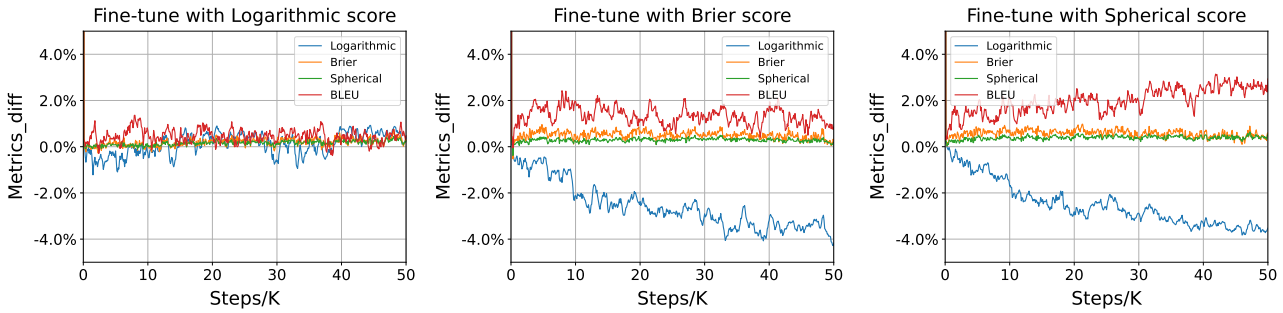

Figure 3. Performance curves on WMT14 En-De test set when fine-tuning with different scoring rules.

Table 6. ROUGE scores of Alpaca fine-tuned large language models on CNN/DailyMail.

Model	RG-1	RG-2	RG-L
LLaMA-7B	28.66	12.49	26.37
LLaMA-7B + Brier	**32.15**	**14.76**	**29.72**
LLaMA-7B + Spherical	30.89	13.87	28.45

Table 7. BLEU scores on the WMT14 En-De test set.

α	1.5	1.75	2	2.25	2.5
α-power	27.91	28.05	28.01	27.84	27.71
pseudo-spherical	28.09	27.91	28.07	27.92	27.64

with loss functions associated with the Brier score and the spherical score. The translation and summarization performance are presented in Table 5 and Table 6, respectively. Due to memory constraints, we only assess the summarization performance of LLaMA-7B. It is surprising to see that fine-tuning with alternative scoring rules can lead to more significant performance improvements on LLMs. Particularly on LLaMA-7B, both scores exhibit an average increase of over 3 BLEU points in translation quality, and the spherical score also demonstrates an average improvement of over 3 ROUGE points in summarization performance.

Figure 2 displays the multi-turn question-answering capabilities of LLMs. Models fine-tuned using the Brier score and the spherical score exhibit stronger overall performance, particularly in extraction and reasoning tasks. In contrast, the model fine-tuned with the logarithmic score is better at writing, roleplay, and STEM tasks.

4.4. Model Dynamics during Fine-tuning

The above experiments show that fine-tuning with other scoring rules can enhance the generative capabilities of language generation models. However, it remains unclear what changes occur within the model during this process. In this section, we investigate the dynamics of the model during the fine-tuning process to better understand its impact. Specifically, on the WMT14 En-De dataset, we pre-train the Transformer using MLE loss and fine-tune it with various scoring rules. Then we track the changes of different scoring rules and also the BLEU score on the test set. Figure 3 illustrates their relative changes, calculated as $\frac{S(p_\theta,q)-S(p_{\theta_{old}},q)}{|S(p_{\theta_{old}},q)|}$.

Table 8. Average BLEU scores on WMT22 test sets.

α	1.5	1.75	2	2.25	2.5
α-power	19.93	20.11	20.93	19.27	19.12
pseudo-spherical	20.42	20.67	20.98	20.03	19.52

As observed, when fine-tuning with the logarithmic score, all metrics fluctuate around their original values since the model is pre-trained with the same score. When fine-tuning with the Brier score or the spherical score, both scores show a certain improvement, accompanied by an increase in BLEU. In contrast, the logarithmic score experiences a significant drop. This interesting phenomenon implies that although different strictly proper scores share the same global optimum, their optimization trajectories might be conflicting, and these scores do not always align with the model's generative capabilities. Therefore, comprehensively considering multiple scores during training can help the model achieve stronger generative capabilities. It also suggests that when assessing language models, a more accurate evaluation could be achieved by considering multiple scores collectively, rather than relying solely on the perplexity.

4.5. Pseudo-spherical Score and Power Score

Previously, we explored the impact of Brier score and spherical score for training language generation models. Here, we further investigate two more general scoring rules, namely the pseudo-spherical score and the α-power score, as described in section 2.4. Both scores include a parameter α, with Brier score and spherical score being their special cases when $\alpha = 2$. To examine the impact of the parameter α, we conduct experiments on both Transformer-base and LLaMA-7B. Table 7 and Table 8 give the results on the WMT14 En-De test set and WMT22 test sets, respectively. Overall, a stable and superior performance is achieved at

Table 9. BLEU scores on the TED bilingual dataset. Avg means the average BLEU. 'Transformer w/ LS' represents a Transformer trained with label smoothing. 'Transformer w/o LS' represents a Transformer trained without smoothing.

X-En	Fr	Ru	He	Ar	It	Nl	Ro	Tr	De	Vi	Avg
Transformer w/o LS	39.39	24.81	37.07	31.79	37.75	35.86	34.40	25.64	34.88	26.48	32.81
+ Brier	40.02	25.43	37.85	32.12	38.41	36.32	35.23	26.17	36.05	26.81	33.44
+ Spherical	40.27	25.49	38.13	32.37	38.67	36.85	36.63	26.43	35.66	27.02	33.75
Transformer w/ LS	40.64	25.74	38.48	32.74	38.87	36.81	35.77	26.80	36.03	27.18	33.91
+ Brier	40.19	25.32	38.36	32.59	38.60	36.40	35.40	26.53	35.65	27.22	33.63
+ Spherical	40.45	25.87	38.40	32.82	38.56	36.68	35.68	26.84	36.00	27.34	33.86

Table 10. Average BLEU scores on TED test sets.

Score	S	S^ϵ	$S^\epsilon_{\log}$
Brier	33.63	33.80	34.49
Spherical	33.86	33.89	34.43

$\alpha = 2$. When $\alpha > 2$, the model performance typically experiences a noticeable decline. In contrast, the models can still maintain a competitive performance when $\alpha < 2$.

4.6. Effect of Score Smoothing

Label smoothing is a commonly used regularization technique for classification networks, particularly crucial in low-resource scenarios. Therefore, we conduct experiments on the TED bilingual dataset to examine whether score smoothing could yield a similar effect. First, we train Transformer models using the smoothed and unsmoothed log-likelihood loss respectively, and then fine-tune them with the unsmoothed Brier and spherical score. The results are presented in Table 9. When not using label smoothing, fine-tuning with alternative scores brings noticeable improvements. However, for models trained with label smoothing, fine-tuning with unsmoothed scores may result in a performance decline, indicating the necessity of score smoothing.

Next, we employ score smoothing techniques to fine-tune the Transformer w/ LS. For simplicity, we only report the average BLEU score in Table 10. The smoothed score S^ϵ results in some improvement, but the impact is relatively minor. By enhancing the smoothing term with the masked logarithmic score, $S^\epsilon_{\log}$ leads to a more noticeable improvement in performance, indicating that score smoothing can also serve as an effective regularization technique.

5. Related Work

Strictly Proper Scoring Rules in Deep Learning. In addition to the widely used logarithmic score, various strictly proper scoring rules have played a significant role in deep learning. The Brier score serves as a training criterion for classification networks (Shoemaker, 1991; Hung et al., 1996; Kline & Berardi, 2005; Hui & Belkin, 2021), as well as an evaluation metric for the quality of uncertainty calibration (Lakshminarayanan et al., 2017; Ovadia et al., 2019; Gruber & Buettner, 2022). The pseudo-spherical score offers solutions for training energy-based models (Yu et al., 2021) and knowledge distillation (Lee & Lee, 2022). In the continuous space, some scoring rules present appealing generative modeling approaches. For example, the Hyvärinen score (Hyvärinen & Dayan, 2005; Ehm & Gneiting, 2012) gives rise to score-based generative models (Song & Ermon, 2019; Song et al., 2021). The energy and kernel score (Gneiting & Raftery, 2007) facilitate the development of generative networks through scoring rule minimization (Gritsenko et al., 2020; Pacchiardi et al., 2021; Pacchiardi & Dutta, 2022).

Loss Functions for Language Generation. Currently, the loss functions used in language generation models are primarily improved versions of cross-entropy loss. One line of research adapts the cross-entropy loss through techniques such as truncation (Kang & Hashimoto, 2020) and reweighting (Ji et al., 2023). Another line of research introduces an additional loss term to complement the cross-entropy loss, such as incorporating reverse cross-entropy (Zhang et al., 2023b), reflective likelihood loss (Dieng et al., 2019), unlikelihood loss (Welleck et al., 2020), and Gaussian prior objective (Li et al., 2020). Stahlberg & Kumar (2022) transforms the multi-class word prediction problem into multiple binary classification problems, which also leads to a well-calibrated model distribution when proper scoring rules (e.g., the logarithmic score used in this work) are employed for binary classification. A recent approach (Shao et al., 2023) involves composing the cross-entropy loss with a convex function, which results in a local but improper scoring rule that alters the shape of the model distribution to be sharper than the data distribution. Other loss functions primarily involve reinforcement learning-based reward optimization, where rewards are derived from evaluation metrics (Ranzato et al., 2016; Shen et al., 2016; Shao et al., 2019; 2021), human feedback (Stiennon et al., 2020; Ouyang et al., 2022), generative adversarial nets (Yu et al., 2017; Yang et al., 2018), or reference demonstrations (Pang & He, 2021). To our knowledge, our work is the first attempt to train lan-

guage generation models using scoring rules other than the logarithmic score.

6. Conclusion

This paper investigates the use of non-local strictly proper scoring rules for training language generation models, with a primary focus on the Brier score and the spherical score. Although these scores do not perform as well as the logarithmic score when training models from scratch, they demonstrate substantial improvements when fine-tuning models that have been pre-trained with the logarithmic score.

Our findings raise several intriguing questions for future research: Despite being strictly proper, different scores still exhibit considerable performance variations when training language models. Are there other scores that exhibit superior performance during pre-training or fine-tuning? What factors contribute to these performance differences, and can we develop additional metrics or properties to determine a score's suitability for training language generation models? Furthermore, beyond model training, it is worth exploring whether these scores can function as evaluation metrics, similar to Perplexity (PPL), for assessing the calibration of language generation models.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. In Leen, T., Dietterich, T., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf.

Bernardo, J. M. Expected information as expected utility. *the Annals of Statistics*, pp. 686–690, 1979.

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bröcker, J. and Smith, L. A. Scoring probabilistic forecasts: The importance of being proper. *Weather and Forecasting*, 22(2):382 – 388, 2007. doi: https://doi.org/10.1175/WAF966.1. URL https://journals.ametsoc.org/view/journals/wefo/22/2/waf966_1.xml.

Dieng, A. B., Cho, K., Blei, D. M., and LeCun, Y. Learning with reflective likelihoods, 2019. URL https://openreview.net/forum?id=SJlh2jR9FX.

Ehm, W. and Gneiting, T. Local proper scoring rules of order two. *The Annals of Statistics*, 40(1):609–637, 2012.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Good, I. J. Rational decisions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 14(1):107–114, 1952.

Gritsenko, A., Salimans, T., van den Berg, R., Snoek, J., and Kalchbrenner, N. A spectral energy distance for parallel speech synthesis. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13062–13072. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/9873eaad153c6c960616c89e54fe155a-Paper.pdf.

Gruber, S. and Buettner, F. Better uncertainty calibration via proper scores for classification and beyond. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 8618–8632. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/3915a87ddac8e8c2f23dbabbcee6eec9-Paper-Conference.pdf.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. Teaching machines to read and comprehend. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/afdec7005cc9f14302cd0474fd0f3c96-Paper.pdf.

Hui, L. and Belkin, M. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=hsFN92eQEla.

Hung, M., Hu, M., Shanker, M., and Patuwo, B. Estimating posterior probabilities in classification problems with neural networks. *International Journal of Computational Intelligence and Organizations*, 1(1):49–60, 1996.

Hyvärinen, A. and Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Ji, H., Ke, P., Hu, Z., Zhang, R., and Huang, M. Tailoring language generation models under total variation distance. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VELL0PlWfc.

Jiao, W., tse Huang, J., Wang, W., Wang, X., Shi, S., and Tu, Z. Parrot: Translating during chat using large language models. *arXiv preprint arXiv:2304.02426*, 2023.

Kang, D. and Hashimoto, T. B. Improved natural language generation via loss truncation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 718–731, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.66. URL https://aclanthology.org/2020.acl-main.66.

Kline, D. and Berardi, V. Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing and Applications*, 14:310–318, 12 2005. doi: 10.1007/s00521-005-0467-y.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Lee, K. and Lee, H. Pseudo-spherical knowledge distillation. In Raedt, L. D. (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 3178–3184. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/441. URL https://doi.org/10.24963/ijcai.2022/441. Main Track.

Li, Z., Wang, R., Chen, K., Utiyama, M., Sumita, E., Zhang, Z., and Zhao, H. Data-dependent gaussian prior objective for language generation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1efxTVYDr.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Liu, Y., Zeng, X., Meng, F., and Zhou, J. Instruction position matters in sequence generation with large language models. *arXiv preprint arXiv:2308.12097*, 2023.

Martins, A. and Astudillo, R. From softmax to sparsemax: A sparse model of attention and multi-label classification. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1614–1623, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/martins16.html.

Martins, P. H., Marinho, Z., and Martins, A. F. T. Sparse text generation. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4252–4273, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.348. URL https://aclanthology.org/2020.emnlp-main.348.

Mikolov, T., Karafiát, M., Burget, L., Černocký, J. H., and Khudanpur, S. Recurrent neural network based language model. In *Interspeech*, 2010. URL https://api.semanticscholar.org/CorpusID:17048224.

Myung, I. J. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Gray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/8558cb408c1d76621371888657d2eb1d-Paper.pdf.

Pacchiardi, L. and Dutta, R. Likelihood-free inference with generative neural networks via scoring rule minimization. *arXiv preprint arXiv:2205.15784*, 2022.

Pacchiardi, L., Adewoyin, R., Dueben, P., and Dutta, R. Probabilistic forecasting with generative networks via scoring rule minimization. *arXiv preprint arXiv:2112.08217*, 2021.

Pang, R. Y. and He, H. Text generation by learning from demonstrations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=RovX-uQ1Hua.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Peters, B., Niculae, V., and Martins, A. F. T. Sparse sequence-to-sequence models. In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1504–1519, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1146. URL https://aclanthology.org/P19-1146.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06732.

Roby, T. B. Belief states: A preliminary empirical study. *Behavioral Sci*, 10(3):255–270, 1965.

Selten, R. Axiomatic characterization of the quadratic scoring rule. *Experimental Economics*, 1:43–61, 1998.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1162. URL https://doi.org/10.18653/v1/p16-1162.

Shao, C., Feng, Y., Zhang, J., Meng, F., Chen, X., and Zhou, J. Retrieving sequential information for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3013–3024, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1288. URL https://www.aclweb.org/anthology/P19-1288.

Shao, C., Feng, Y., Zhang, J., Meng, F., and Zhou, J. Sequence-Level Training for Non-Autoregressive Neural Machine Translation. *Computational Linguistics*, pp. 1–35, 10 2021. ISSN 0891-2017. doi: 10.1162/coli_a_00421. URL https://doi.org/10.1162/coli_a_00421.

Shao, C., Ma, Z., Zhang, M., and Feng, Y. Beyond mle: Convex learning for text generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1683–1692, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1159. URL https://aclanthology.org/P16-1159.

Shoemaker, P. A note on least-squares learning procedures and classification by neural network models. *IEEE Transactions on Neural Networks*, 2(1):158–160, 1991. doi: 10.1109/72.80304.

Shuford Jr, E. H., Albert, A., and Edward Massengill, H. Admissible probability measurement procedures. *Psychometrika*, 31(2):125–145, 1966.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Stahlberg, F. and Kumar, S. Jam or cream first? modeling ambiguity in neural machine translation with SCONES. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V. (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4950–4961, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.365. URL https://aclanthology.org/2022.naacl-main.365.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJeYe0NtvH.

Xu, J., Zhou, H., Gan, C., Zheng, Z., and Li, L. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7361–7373, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.571. URL https://aclanthology.org/2021.acl-long.571.

Yang, Z., Chen, W., Wang, F., and Xu, B. Improving neural machine translation with conditional sequence generative adversarial nets. In Walker, M., Ji, H., and Stent, A. (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1346–1355, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1122. URL https://aclanthology.org/N18-1122.

Yu, L., Zhang, W., Wang, J., and Yu, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 2852–2858. AAAI Press, 2017.

Yu, L., Song, J., Song, Y., and Ermon, S. Pseudo-spherical contrastive divergence. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22348–22362. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/bc5fcb0018cecacba559dc512740091b-Paper.pdf.

Zeng, J., Meng, F., Yin, Y., and Zhou, J. Tim: Teaching large language models to translate with comparison. *arXiv preprint arXiv:2307.04408*, 2023.

Zhang, S., Fang, Q., Zhang, Z., Ma, Z., Zhou, Y., Huang, L., Bu, M., Gui, S., Chen, Y., Chen, X., and Feng, Y. Bayling: Bridging cross-lingual alignment and instruction following through interactive translation for large language models. *arXiv preprint arXiv:2306.10968*, 2023a.

Zhang, S., Wu, S., Irsoy, O., Lu, S., Bansal, M., Dredze, M., and Rosenberg, D. MixCE: Training autoregressive language models by mixing forward and reverse cross-entropies. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9027–9050, Toronto, Canada, July

2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.502. URL https://aclanthology.org/2023.acl-long.502.

A. α-power Score and α-entmax Loss

We are grateful for the very insightful comments provided by Reviewer gaCL, which motivated us to investigate the connection between the α-power score and α-entmax loss.

Softmax has a limitation in that it cannot produce probabilities exactly equal to zero. To generate sparse probability distributions, methods such as sparsemax (Martins & Astudillo, 2016) and α-entmax (Peters et al., 2019; Martins et al., 2020) have been proposed, where sparsemax is a special case of α-entmax with $\alpha = 2$. Given the probability space $\triangle^d = \{p \in \mathbb{R}^d \colon p \geq 0, \|p\|_1 = 1\}$, α-entmax is a transformation $\mathbb{R}^d \to \triangle^d$, defined as:

$$\alpha\text{-entmax}(z) = \arg\max_{p \in \triangle^d} p^\top z + H_\alpha^\top(p), \tag{23}$$

where $H_\alpha^\top(p)$ is a family of entropies parametrized by a scalar $\alpha \geq 1$, known as Tsallis α-entropies:

$$H_\alpha^\top(p) = \begin{cases} \frac{1}{\alpha(\alpha-1)} \sum_j (p_j - p_j^\alpha), & \alpha > 1 \\ -\sum_j p_j \log p_j, & \alpha = 1 \end{cases}. \tag{24}$$

The associated loss function is called α-entmax loss:

$$\mathcal{L}_\alpha(z, x) = (p - e_x)^\top z + H_\alpha^\top(p), \tag{25}$$

where $p = \alpha\text{-entmax}(z)$ and e_x is the one-hot vector corresponding to the ground truth word x. We will show that under certain conditions (i.e., the probability of ground truth word $p_x > 0$, $\alpha > 1$), the α-entmax loss is equivalent to the following token-level loss based on α-power score:

$$\mathcal{L}_{\alpha\text{-power}}(p, x) = (\alpha - 1) \sum_{j=1}^{m} p_j^\alpha - \alpha p_x^{\alpha-1}. \tag{26}$$

To solve the constrained problem in equation 23, we can apply the Lagrange multiplier:

$$f(p, \lambda, \mu) = p^\top z + H_\alpha^\top(p) - \lambda(\sum_j p_j - 1) - \sum_j \mu_j p_j. \tag{27}$$

$$\frac{\partial f(p, \lambda, \mu)}{\partial p_j} = z_j - \frac{1}{\alpha - 1} p_j^{\alpha-1} - \lambda - \mu_j = 0. \tag{28}$$

Due to the complementary slackness condition of the KKT, if the solution $p_j > 0$, then we have $\mu_j = 0$, which yields:

$$z_j = \lambda + \frac{p_j^{\alpha-1}}{\alpha - 1}. \tag{29}$$

Similarly, if the probability of ground truth word $p_x > 0$, then $z_x = \lambda + \frac{p_x^{\alpha-1}}{\alpha-1}$. Substituting these into equation 25, we obtain:

$$\begin{aligned} \mathcal{L}_\alpha(z, x) &= \sum_{j, p_j > 0} p_j z_j - z_x + H_\alpha^\top(p) = \sum_{j, p_j > 0} p_j(\lambda + \frac{1}{\alpha - 1} p_j^{\alpha-1}) - \lambda - \frac{p_x^{\alpha-1}}{\alpha - 1} + H_\alpha^\top(p) \\ &= \sum_j \frac{p_j^\alpha}{\alpha - 1} - \frac{p_x^{\alpha-1}}{\alpha - 1} + \frac{1}{\alpha(\alpha - 1)} - \sum_j \frac{p_j^\alpha}{\alpha(\alpha - 1)} \\ &= \frac{1}{\alpha(\alpha - 1)} [(\alpha - 1) \sum_j p_j^\alpha - \alpha p_x^{\alpha-1} + 1] \\ &= \frac{\mathcal{L}_{\alpha\text{-power}}(p, x) + 1}{\alpha(\alpha - 1)}. \end{aligned} \tag{30}$$

As shown, the α-entmax loss is a linear transformation of the α-power score based loss, so they are fundamentally equivalent. This reveals the propriety of the α-entmax loss. However, the equivalence does not hold when $p_x = 0$. In this case, z_x falls below the threshold of obtaining positive probability, causing the gradient from the probability vector $\frac{\partial p}{\partial z_x}$ to be 0. This makes it theoretically impossible to obtain a gradient from probability-based loss functions. Therefore, when applying other strictly proper scoring rules to the training of sparse transformations, adjustments are still necessary to ensure that the gradient can be transmitted to the golden logit z_x.

B. Scoring Rules as Beam Search Objective

We are grateful for the very insightful comments provided by Reviewer z8jq, which inspired us to investigate the application of scoring rules as objectives for beam search.

In the realm of conditional generation tasks such as machine translation and text summarization, beam search is a widely adopted decoding strategy aimed at finding the output sequence y with the highest length-normalized log-probability. The formal objective of beam search can be expressed as:

$$\max_{y} \frac{\sum_{t=1}^{|y|} \log p_\theta(y_t|x, y_{<t})}{|y|^\alpha}, \tag{31}$$

where α denotes the length penalty hyperparameter. The above equation can also be understood as maximizing the sum of token-level logarithmic scores. Similarly, we can consider having beam search optimize other token-level scoring rules:

$$\max_{y} \frac{\sum_{t=1}^{|y|} S(p_\theta(\cdot|x, y_{<t}), y_t)}{|y|^\alpha}. \tag{32}$$

Here, S can be the Brier score $S(p, i) = 2p_i - \sum_{j=1}^{m} p_j^2$, the spherical score $S(p, i) = \frac{p_i}{|p|}$, or other strictly proper scoring rules. A critical aspect is the sign (positive or negative) of the scoring rule. Given their definitions, the logarithmic score is inherently negative, the spherical score is positive, and the sign of the Brier score is uncertain. For a negative score like the logarithmic score, models tend to favor shorter sentences, whereas the length penalty α can counterbalance this by encouraging longer output. Conversely, for a positive score like the spherical score, models are inclined to generate longer sentences, and here, the length penalty α serves to encourage shorter sentences. To unify them, we subtract 1 from both the Brier score and the spherical score to ensure they are non-positive:

$$S'_{Brier} = 2p_i - \sum_{j=1}^{m} p_j^2 - 1 \leq 2p_i - p_i - 1 \leq 0, \quad S'_{Spherical} = \frac{p_i}{|p|} - 1 \leq 1 - 1 = 0. \tag{33}$$

We conduct experiments on the WMT14 En-De dataset to evaluate the impact of different scoring rules on the quality of generated text when used as the objective for beam search. The results are presented in Table 11. The results indicate that, among the three scoring rules examined, the logarithmic score yields the best performance, with the Brier score outperforming the spherical score. However, there are exceptions. For instance, the model fine-tuned with the spherical score demonstrated a preference for beam search optimization using the spherical score over the Brier score.

Table 11. BLEU scores on WMT14 En-De when applying different scoring rules as beam search objective.

Model	**Logarithmic**	**Brier**	**Spherical**
Transformer	27.61	27.56	27.23
Transformer + Brier	28.01	27.95	27.53
Transformer + Spherical	28.07	27.40	27.78

Our investigation into the use of different scoring rules as objectives for beam search is far from exhaustive. It is plausible that other strictly proper scoring rules could surpass the performance of logarithmic score. We leave this for future exploration.