Journal of Machine Learning Research 24 (2023) 1-92 Submitted 1/23; Published 10/23

MAUVE Scores for Generative Models: Theory and Practice

Krishna Pillutla[1*] pillutla@cs.washington.edu
Lang Liu[2*] liu16@uw.edu
John Thickstun[3] jthickstun@stanford.edu
Sean Welleck[1,4] wellecks@cs.washington.edu
Swabha Swayamdipta[5] swabhas@usc.edu
Rowan Zellers[6] rowanz@cs.washington.edu
Sewoong Oh[1,7] sewoong@cs.washington.edu
Yejin Choi[4,7] yejin@cs.washington.edu
Zaid Harchaoui[2] zaid@uw.edu
[1] *Google Research*
[2] *Department of Statistics, University of Washington*
[3] *Department of Computer Science, Stanford University*
[4] *Allen Institute for Artificial Intelligence*
[5] *Viterbi School of Engineering, University of Southern California*
[6] *OpenAI*
[7] *Paul G. Allen School of Computer Science and Engineering, University of Washington*

Editor: Kilian Weinberger

Abstract

Generative artificial intelligence has made significant strides, producing text indistinguishable from human prose and remarkably photorealistic images. Automatically measuring how close the generated data distribution is to the target distribution is central to diagnosing existing models and developing better ones. We present Mauve, a family of comparison measures between pairs of distributions such as those encountered in the generative modeling of text or images. These scores are statistical summaries of divergence frontiers capturing two types of errors in generative modeling. We explore three approaches to statistically estimate these scores: vector quantization, non-parametric estimation, and classifier-based estimation. We provide statistical bounds for the vector quantization approach.

Empirically, we find that the proposed scores paired with a range of f-divergences and statistical estimation methods can quantify the gaps between the distributions of human-written text and those of modern neural language models by correlating with human judgments and identifying known properties of the generated texts. We demonstrate in the vision domain that Mauve can identify known properties of generated images on par with or better than existing metrics. In conclusion, we present practical recommendations for using Mauve effectively with language and image modalities.

. *These authors contributed equally to this work.

Keywords: Generative models, evaluation, divergence frontiers, neural text generation, large language models, f-divergences, statistical estimation

1. Introduction

Large-scale generative artificial intelligence models show an ability to produce human-like text and realistic images. Recent chatbots such as ChatGPT/GPT-4 (OpenAI, 2023), Bard (Google, 2023), and Ernie Bot (Sun et al., 2021) have rapidly gained wide prominence in the general public for their articulate responses across many topics and styles. More generally, large language models such as Llama-2 (Touvron et al., 2023), Falcon (Almazrouei et al., 2023), Bloom (Workshop, 2022), and Mistral (Jiang et al., 2023), as well as image and multi-modal generative models such as Stable Diffusion (Rombach et al., 2022), Imagen (Saharia et al., 2022), and CM3leon (Yu et al., 2023) can produce original content in response to queries in the form of blog posts, poetry, computer programs, and artwork.

However, evaluating the distributions captured by such large-scale generative models requires substantial effort. Automatic measures can dramatically reduce the cost of evaluation, in turn making it easier to rapidly develop models, choose hyperparameters, and understand a model's capabilities.

One approach to evaluation is to compare a generative model's distribution Q with the target distribution P of the real data that it aims to model. Doing so requires considering two types of errors: (I) the mass of Q that has a low probability under P where the model produces unrealistic or degenerate data, and (II) the mass of P that has a low probability under Q where the model is not able to produce some class of realistic data. However, quantifying these errors in a principled, computationally tractable manner is challenging when faced with real-world text or image distributions.

We present a family of comparison measures between pairs of probability distributions, such as those encountered in the generative modeling of text and images. Building upon the notion of divergence frontiers proposed by Djolonga et al. (2020), our measures are statistical summaries of *f-divergence frontiers*, which capture the two types of errors. We explore three methods for estimating these divergence frontiers and their scalar summaries. We provide statistical bounds for two of these estimation methods—vector quantization and nearest-neighbor estimation—as well as theoretical guidance on choosing the level of vector quantization. In the spirit of popular metrics in natural language processing such as Bleu (Papineni et al., 2002) and Rouge (Lin, 2004), we call these measures Mauve scores.

We develop the scores in practice for open-ended text generation. We find that, for a range of f-divergences and estimation methods, these measures quantify the gap between the distributions of human-written text and those of modern neural language models efficiently and robustly. Moreover, we show that these measures extend to image distributions, aligning well with the widely used Fréchet distance in the computer vision domain in quantifying the effect of sampling algorithms and architectural improvements. Together, our theoretical and empirical analyses demonstrate that Mauve provides a principled, effective, and powerful recipe for comparing distributions of complex high-dimensional text and images.

1.1 Contributions

We make the following contributions in this work.

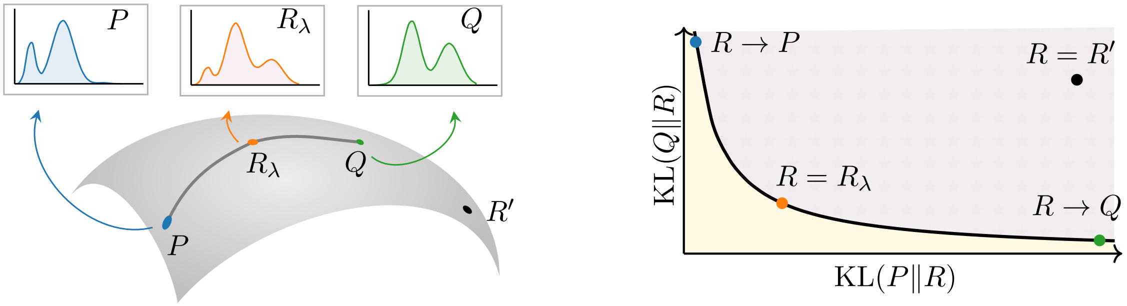


Figure 1: Left: Comparing two distributions P and Q. Here, $R_\lambda = \lambda P + (1-\lambda)Q$ is the interpolation between P and Q for $\lambda \in (0,1)$ and R' denotes some arbitrary distribution. **Right**: The corresponding divergence frontier (black curve) between P and Q. The interpolations R_λ for $\lambda \in (0,1)$ make up the frontier, while all other distributions such as R' must lie above the frontier.

Statistical Summaries of Divergence Frontiers (Section 3). Our goal is to provide a scalar summary of the discrepancy between a generative model Q and the target distribution P that it aims to model. To do so, following Djolonga et al. (2020), we consider two types of costs: (I) the mass of Q that has low probability under P, and (II) the mass of P that has low probability under Q. We formalize these costs using a *divergence frontier*,

$$\mathcal{F}_f(P,Q) = \Big\{ \big(D_f(P\|R_\lambda),\, D_f(Q\|R_\lambda)\big) \,:\, \lambda \in (0,1) \Big\},$$

where $R_\lambda = \lambda P + (1-\lambda)Q$, and D_f is an f-divergence such as the Kullback–Leibler (KL) divergence. See Figure 1 for an illustration. This extends the frontiers of (Djolonga et al., 2020) to general f-divergences. We shall show in Section 3 that the nice properties of the divergence frontiers also extend to their variants based on f-divergences.

We propose three scalar statistical summaries of divergence frontiers. The first summary measures the area under a transformed divergence frontier:

$$\textsc{Mauve}_f(P,Q) = \mathrm{AUC}\left(\big\{ \big(\exp(-x),\, \exp(-y)\big) \,:\, (x,y) \in \mathcal{F}_f(P,Q) \big\} \cup \{(1,0),(0,1)\} \right) .$$

Here, $\exp(\cdot)$ monotonically transforms the frontier to account for unbounded divergences.

Second, we consider an integral summary that sweeps over the coordinates on the divergence frontier and accumulates their costs:

$$\mathrm{FI}_f(P,Q) := 2\int_0^1 \big(\lambda\, D_f(P\|R_\lambda) + (1-\lambda) D_f(Q\|R_\lambda)\big)\, \mathrm{d}\lambda.$$

Finally, the third summary simply uses costs from the mid-point of the frontier, i.e., the coordinates corresponding to $\lambda = 1/2$:

$$\mathrm{Mid}_f(P,Q) := \frac{1}{2}\, D_f(P\|R_{1/2}) + \frac{1}{2} D_f(Q\|R_{1/2})\, .$$

At their core, all three summaries are based on f-divergences. Thus, all three benefit from our estimation algorithms and error bounds for f-divergences, which we discuss next.

Statistical Estimation Algorithms (Section 4). We give algorithms for computing the summaries $\textsc{Mauve}_f$, FI_f, and Mid_f on real-world distributions of text or images. This requires computing f-divergences between the target distribution P and the model distribution Q, which is challenging due to the lack of direct access to P and Q, and the large support of each distribution. To address these challenges, we propose three methods for estimating divergence frontiers from i.i.d. samples using embeddings of the data (e.g., from a large language model for text data):

1. *Quantization*: we jointly quantize the distributions P and Q in some embedding space to form two multinomial distributions, then estimate the divergence frontier between the two multinomial distributions.

2. *Nearest-neighbor*: we use the nearest neighbors (in some embedding space) of each sample to estimate the likelihood ratio $P(x)/Q(x)$, which we use to estimate the required f-divergences.

3. *Classifier*: we train a classifier to identify whether each sample belongs to the target or model distribution. We use the classifier to estimate the likelihood ratio and, in turn, the required f-divergences.

Error Bounds. We develop error bounds for the first quantization approach. The total estimation error of the divergence frontier consists of two parts: (i) the statistical error in estimating the frontier from samples, and (ii) the quantization error that arises from passing from the original distributions to their quantized versions.

For the statistical error, Theorem 10 gives an error bound that allows for long tails and countable support of the distribution P. This improves over a naive bound that does not allow for distributions with long tails, and requires finite support. A key technique that enables this result is considering the *missing mass* (Good, 1953): the total probability that does not appear in the finite sample used to estimate the frontier. When the two distributions P and Q intersect on a finite set of k elements, the bounds simplify further. For example, we give the following statistical error bound on the integral summary (Eq. 12):

$$\mathbb{E}|\mathrm{FI}(\hat{P}_n, \hat{Q}_n) - \mathrm{FI}(P, Q)| \le \tilde{O}\left(\sqrt{\frac{k}{n}} + \frac{k}{n}\right) ,$$

where $\hat{P}_n$ and $\hat{Q}_n$ are the empirical estimators and n is the number of samples. We give a similar bound for general f-divergences (Eq. 11). Our results hold under assumptions that are satisfied by many common f-divergences (Table 9). To improve the statistical performance of empirical estimators when the quantization size k is large, we also apply *add-constant* smoothing to estimate the two distributions—we add a small constant $b > 0$ to the counts of each bin and normalize them to form a distribution. We prove in Theorem 12 a statistical error bound for the add-constant estimators. Applied to the integral summary, the bound is (Eq. 17)

$$\mathbb{E}|\mathrm{FI}(\hat{P}_n^b, \hat{Q}_n^b) - \mathrm{FI}(P, Q)| \le \tilde{O}\left(\frac{\sqrt{kn} + kb}{n + kb}\right) ,$$

where $\hat{P}_n^b$ and $\hat{Q}_n^b$ are the add-constant estimators. A similar bound for general f-divergences is given in Eq. 16.

For the quantization error, we show that there exists a quantization scheme with error $O(1/k)$, where k is the size of the k-partition used to quantize the sample space. Our analysis is inspired by the asymptotic approximation of an f-divergence with increasingly finer partitions (Györfi and Nemetz, 1978, Theorem 6). Combining the statistical and quantization error bounds gives us a bound on the total error of the integral summary (Eq. 20):

$$\mathbb{E}|\mathrm{FI}(\hat{P}_{\mathcal{S}_k,n}, \hat{Q}_{\mathcal{S}_k,n}) - \mathrm{FI}(P, Q)| \le \tilde{O}\left(\sqrt{\frac{k}{n}} + \frac{k}{n} + \frac{1}{k}\right).$$

We discuss how to operationalize the nonparametric nearest-neighbor estimation with dimensionality reduction via principal component analysis (PCA). For nearest-neighbor estimation, we discuss bounds from Noshad et al. (2017) (Theorem 17).

Experiments (Section 7). Our experiments are organized into multiple parts, mainly focusing on the open-ended text generation setting.

We start by analyzing the effectiveness of the proposed measure for comparing text distributions. We focus on the area summary using the KL divergence computed with vector quantization. We demonstrate that the proposed measures correlate with human quality judgments (Section 7.1) and quantify known properties of generated text (Section 7.2). The main focus of the rest of the experimental study is to analyze the effects of each of the components of the evaluation pipeline: the estimation method, the choice of the divergence, and the choice of the embedding.

First, we consider different **estimation methods**: vector quantization, nearest neighbor estimation, and classifier-based estimation (Section 7.3). We also consider a popular parametric Gaussian approximation method—assuming that embedded samples from the target and model distributions are distributed according to multivariate Gaussians, we estimate the parameters of each Gaussian and estimate the divergence frontier by numerical integration (see Appendix C for more details). We find that all estimation methods identify expected quality trends and correlate with human evaluations. However, nearest-neighbor and classifier-based estimation show a slightly decreased ability to identify good hyperparameter values, while parametric estimation requires extreme dimensionality reduction. Thus, we recommend vector quantization as a default.

Second, we experiment with other ***f*-divergences and optimal transport costs** (Section 7.4). Specifically, we compare different variants of the proposed measure based on (i) alternate f-divergences, (ii) other statistical summaries of the divergence frontier, and (iii) summaries of frontiers based on optimal transport distances. We find that all the quantities based on f-divergences correlate perfectly. On the other hand, some of the optimal transport distances fail to capture expected trends. These results demonstrate the flexibility and effectiveness of our proposed measures.

Third, we perform a thorough exploration of the **effect of the embedding** in the evaluation pipeline (Section 7.5). Our experiments reveal that the embedding is crucial to the empirical success of Mauve. While most large language model embeddings (either a masked or a causal language model, including the model used to generate the text) and even shallow GloVe (Pennington et al., 2014) embeddings yield useful comparison measures, we find that string kernel-based embeddings or embedding-free direct estimation methods fail to capture expected trends.

Finally, we demonstrate that our measures generalize to other AI domains beyond text. Specifically, we show that in the **image domain**, our measure recovers expected trends with respect to the sampling algorithm and model size, and correlates perfectly with the widely used Fréchet distance in this setting (Section 7.7).

Previous Papers. This work builds upon two previous shorter conference papers. The first (Pillutla et al., 2021) introduces the area summary in the context of open-ended text generation and conducts an empirical study. The second (Liu et al., 2021) studies the statistical theory behind estimating divergence frontiers with vector quantization and smoothed distribution estimators. This work unifies both of these works and makes several further contributions.

First, we introduce the notion of f-divergence frontiers and three scalar summaries, generalizing the area summary from (Pillutla et al., 2021) and the integral summary from (Liu et al., 2021). We also systematically study the properties of the three summaries (Section 3). Second, we consider three estimation algorithms (Section 4), based on nonparametric estimation, classifier-based estimation, and a parametric Gaussian approximation, and empirically compare their performance for open-ended text generation (Section 7.3). Empirically, we perform a thorough exploration of alternatives based on f-divergences and optimal transport (Section 7.4). We also probe the effect of the embedding (Section 7.5), and perform experiments in the vision domain (Section 7.7), not covered in the previous two papers.

2. Background and Setup

We discuss the basics of open-ended text generation and set up the problem of comparing multiple generative models.

2.1 Language Modeling and Open-Ended Text Generation

We start with neural autoregressive language models since these form the backbone of prevailing approaches to text generation.

Language Modeling. Consider a sequence $\boldsymbol{x} = (x_1, \cdots, x_{|\boldsymbol{x}|})$ of natural language text, where each x_i belongs to a finite vocabulary V (e.g., characters or words). An autoregressive language model $\hat{P}(\cdot \,|\, \boldsymbol{x}_{1:t})$ models the conditional distribution over the next token x_{t+1} following the sequence $\boldsymbol{x}_{1:t}$. While neural language models, i.e., language models parameterized by a neural network, date back to at least (Bengio et al., 2003; Collobert et al., 2011), contemporary models are based on the transformer architecture (Vaswani et al., 2017) summarized in Figure 2 (left).

The usual training objective for neural language modeling is via supervised multi-class classification of the next token. We assume that there is an underlying distribution $P(\cdot \,|\, \boldsymbol{x}_{1:t})$ for the next token x_{t+1} humans would write in continuation to a prefix $\boldsymbol{x}_{1:t}$. The training procedure aims to minimize the Kullback-Liebler (KL) divergence between the distributions $P(\cdot \,|\, \boldsymbol{x}_{1:t})$ and $\hat{P}(\cdot \,|\, \boldsymbol{x}_{1:t})$ assigned by humans and the language model respectively over the next token x_{t+1} in continuation to a context $\boldsymbol{x}_{1:t} \sim P_t$ coming from *human-written* text:

$$\min_{\theta} \; \mathbb{E}_{t \sim \mathrm{Unif}([T-1])} \mathbb{E}_{\boldsymbol{x}_{1:t} \sim P_t} \left[\mathrm{KL}\Big(P(\cdot \,|\, \boldsymbol{x}_{1:t}) \Big\| \hat{P}_\theta(\cdot \,|\, \boldsymbol{x}_{1:t}) \Big) \right] \,, \tag{1}$$

where T is the maximum sequence length. Since neither the distribution P_t over prefixes of length t nor the distribution $P(\cdot \,|\, \boldsymbol{x}_{1:t})$ over the next token is known in practice, plug-in estimates of both are employed in practice.

Autoregressive models also yield an estimate of the joint probability $\hat{P}(\boldsymbol{x})$ of a sequence $\boldsymbol{x} = (x_1, \cdots, x_{|\boldsymbol{x}|})$ as

$$\hat{P}(\boldsymbol{x}) = \prod_{t=0}^{|\boldsymbol{x}|-1} \hat{P}(x_{t+1} \,|\, \boldsymbol{x}_{1:t}) \,.$$

Open-Ended Text Generation. The open-ended text generation task asks us to output text $\hat{\boldsymbol{x}}_{s+1:|\hat{\boldsymbol{x}}|}$ in continuation of a given context $\boldsymbol{x}_{1:s}$. In contrast to directed text generation tasks such as translation, summarization, and question-answering, the task here is open-ended in that the context size $s \ll |\hat{\boldsymbol{x}}|$ is typically small and does not meaningfully constrain the output space. Unlike directed text generation tasks such as translation, summarization, and question-answering, the goal here is to generate text that is coherent, fluent, creative, and engaging. Since these criteria are hard to make mathematically precise, we instead consider the surrogate goal of generating text which is *human-like*, such that generated text samples can pass for samples from the distribution P over human written text sequences.

We model a text generation system as a probability distribution $Q(\cdot \,|\, \boldsymbol{x}_{1:s})$ such that its generated text $\hat{\boldsymbol{x}}_{s+1:|\hat{\boldsymbol{x}}|}$ is an i.i.d. sample from Q. Given a neural autoregressive language model $\hat{P}$, we can generate open-ended text in a serial, left-to-right fashion, by sampling $\hat{x}_{s+1} \sim \hat{P}(\cdot|\boldsymbol{x}_{1:s})$, $\hat{x}_{s+2} \sim \hat{P}(\cdot|\boldsymbol{x}_{1:s}, \hat{x}_{s+1})$, etc. This is also known as *ancestral sampling*, and the induced distribution Q over sequences is

$$Q_{\mathrm{samp}}(\boldsymbol{x}_{1:s}, \hat{\boldsymbol{x}}_{s+1:|\hat{\boldsymbol{x}}|}) = \prod_{t=1}^{s} P(x_t|\boldsymbol{x}_{1:t-1}) \prod_{t=s+1}^{|\hat{\boldsymbol{x}}|} \hat{P}(\hat{x}_t|\boldsymbol{x}_{1:s}, \hat{\boldsymbol{x}}_{s+1:t-1}) \,,$$

where we assume that the prefix $\boldsymbol{x}_{1:s} \sim P_s$ is drawn from the human distribution. Note that the distribution Q_{samp} is identical to $\hat{P}$, expect for the prefix $\boldsymbol{x}_{1:s}$. General decoding algorithms produce samples from a reshaped model distribution, as we discuss next.

Decoding Algorithms. Assuming the language model learning has succeeded, we have that $\hat{P}(\cdot \,|\, \boldsymbol{x}_{1:t}) \approx P(\cdot \,|\, \boldsymbol{x}_{1:t})$ for prefixes $\boldsymbol{x}_{1:t} \sim P_t$ drawn from the distribution of human-written text, in the sense that the objective of (1) is bounded above by some $\varepsilon > 0$. However, for $\hat{\boldsymbol{x}}_{1:t}$ drawn from a distribution Q_t which is different from the human distribution P_t, the model's next-token distribution $\hat{P}(\cdot \,|\, \hat{\boldsymbol{x}}_{1:t})$ can be quite different from $P(\cdot \,|\, \hat{\boldsymbol{x}}_{1:t})$ of humans. In the iterative process of ancestral sampling, the gap between $P(\hat{\boldsymbol{x}}_{1:t})$ and $Q_{\mathrm{samp}}(\hat{\boldsymbol{x}}_{1:t})$ keep increasing as the generation length t grows larger, so that Q_{samp} is quite far from P. This leads to *decoding algorithms* which produce samples

$$\hat{x}_{t+1} \sim Q(\cdot \,|\, \boldsymbol{x}_{1:s}, \hat{\boldsymbol{x}}_{s+1:t}) \,,$$

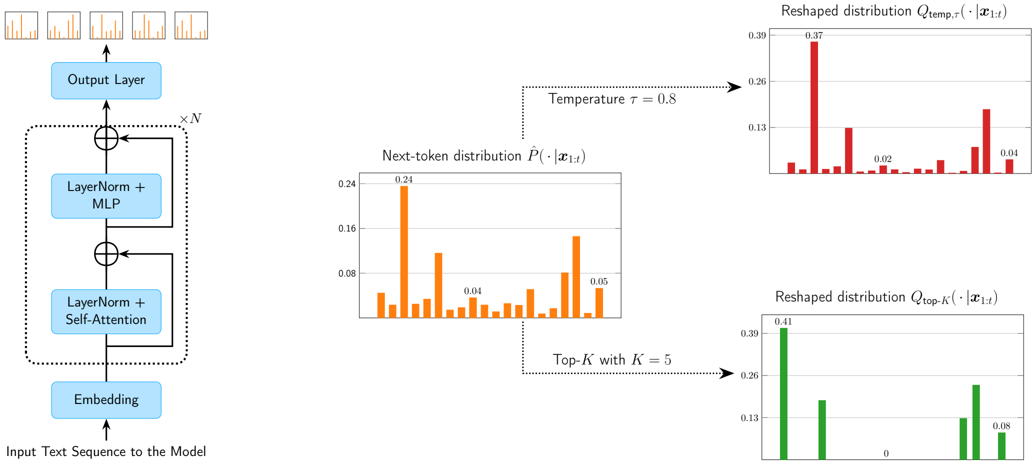

Output Layer
×N
LayerNorm + MLP
LayerNorm + Self-Attention
Embedding
Input Text Sequence to the Model
Next-token distribution $\hat{P}(\cdot\,|\boldsymbol{x}_{1:t})$
0.24
0.16
0.08
0.24
0.04
0.05
Temperature $\tau = 0.8$
Reshaped distribution $Q_{\text{temp},\tau}(\cdot\,|\boldsymbol{x}_{1:t})$
0.39
0.26
0.13
0.37
0.02
0.04
Top-K with $K = 5$
Reshaped distribution $Q_{\text{top-}K}(\cdot\,|\boldsymbol{x}_{1:t})$
0.39
0.26
0.13
0.41
0
0.08

Figure 2: Left: The transformer architecture takes in a text sequence $\boldsymbol{x} = (x_1, \ldots, x_{|\boldsymbol{x}|})$ and outputs the next-token distribution $\hat{P}(\cdot\,|\boldsymbol{x}_{1:t})$ for each prefix $\boldsymbol{x}_{1:t}$. **Right**: Illustration of how decoding algorithms (specifically, temperature rescaling and top-K decoding) reshape the model's next-token distribution.

where $Q(\cdot\,|\,\boldsymbol{x}_{1:t})$ is a reshaping of the language model $\hat{P}(\cdot\,|\,\boldsymbol{x}_{1:t})$ in order to promote more conservative outputs. We now define a few popular decoding algorithms; see also Figure 2 (right) for an illustration.

Temperature rescaling (Ackley et al., 1985) applies to language models parameterized with a softmax function:

$$\hat{P}(x_{t+1}\,|\,\boldsymbol{x}_{1:t}) = \frac{\exp\big(\phi(x_{t+1}|\boldsymbol{x}_{1:t})\big)}{\sum_{x\in V}\exp\big(\phi(x|\boldsymbol{x}_{1:t})\big)}\,,$$

for some unnormalized scoring function $\phi(\cdot\,|\,\boldsymbol{x}_{1:t}) : V \to \mathbb{R}$. This decoding algorithm rescales the term inside the exponential with a "temperature" parameter $\tau > 0$:

$$Q_{\text{temp},\tau}(x_{t+1}\,|\,\boldsymbol{x}_{1:t}) = \frac{\exp\big(\frac{1}{\tau}\phi(x_{t+1}|\boldsymbol{x}_{1:t})\big)}{\sum_{x'_{t+1}\in V}\exp\big(\frac{1}{\tau}\phi(x'_{t+1}|\boldsymbol{x}_{1:t})\big)}\,.$$

When $\tau < 1$, the distribution $Q_{\text{temp},\tau}(\cdot\,|\,\boldsymbol{x}_{1:t})$ becomes more peaked around the most likely next tokens, making the distribution more conservative.

For an integer $K < |V|$, *top-K sampling* (Fan et al., 2018) applies the transformation

$$Q_{\text{top-}K}(x_{t+1}|\boldsymbol{x}_{1:t}) = \begin{cases} \frac{1}{Z}\,\hat{P}(x_{t+1}|\boldsymbol{x}_{1:t})\,, & \text{if } x_{t+1} \in V_{\text{top-}K}\,, \\ 0\,, & \text{else,} \end{cases}$$

where Z is a normalizing constant, and $V_{\text{top-}K} = \{z_{(1)}, \cdots, z_{(K)}\} \subset V$ is the set of the K highest scoring tokens satisfying

$$\hat{P}(z_{(1)}|\boldsymbol{x}_{1:t}) \ge \cdots \ge \hat{P}(z_{(K)}|\boldsymbol{x}_{1:t}) \ge \max_{z\in V\setminus V_{\text{top-}K}} \hat{P}(z|\boldsymbol{x}_{1:t})\,.$$

The extreme $K = |V|$ corresponds to ancestral sampling. The other extreme $K = 1$ is known as *greedy decoding*, which corresponds to choosing the most likely next token iteratively. Greedy decoding is often used to approximate the most likely sequence $\arg\max_{\boldsymbol{x}} P(\boldsymbol{x}|\boldsymbol{x}_{1:t})$.

Nucleus sampling (Holtzman et al., 2020), similar to top-K sampling, returns a sparse distribution. Given a parameter $p \in (0, 1)$, it applies the transformation

$$Q_{\mathrm{nuc},p}(x_{t+1} \mid \boldsymbol{x}_{1:t}) = \begin{cases} \frac{1}{Z}\,\hat{P}(x_{t+1} \mid \boldsymbol{x}_{1:t}), & \text{if } x_{t+1} \in V_{\mathrm{nuc},p}, \\ 0, & \text{else}, \end{cases} \tag{2}$$

where Z is again a normalizing constant. Here, the top-p vocabulary $V_{\mathrm{nuc},p}$ is the smallest set $V' \subset V$ such that $\sum_{x \in V'} \hat{P}(x|\boldsymbol{x}_{1:t}) \ge p$.

2.2 Comparing Generative Models

The usual approach to evaluating a text generation model is to compare the output of the model to human-written text for the same prompt (Papineni et al., 2002; Lin, 2004, etc.). This paradigm, however, breaks down for open-ended generation since there can be multiple correct outputs.

We frame the problem as comparing two distributions. Let $Q \in \mathcal{P}(\mathcal{X})$ denote the model distribution over some data space $\mathcal{X}$ such as text sequences or images and let $P \in \mathcal{P}(\mathcal{X})$ denote the target real data distribution. For text distributions, Q depends on the underlying language model $\hat{P}$ as well as the decoding algorithm. The goal of open-ended text generation is to generate human-like text and the goal of image generation is to generate photorealistic images. Both these goals can be framed as finding a model distribution Q that is as close to P as possible in some metric. Therefore, we cast the evaluation of the generative model as measuring the gap between the model distribution Q and the target distribution P. We will make this precise in Section 3.

2.3 Information Divergences

We review the definition of f-divergences and give a few examples.

Definition 1. *Let $f : (0, \infty) \to \mathbb{R}_+$ be a convex function with $f(1) = 0$. Let $P, Q \in \mathcal{P}(\mathcal{X})$ be dominated by some measure $\mu \in \mathcal{P}(\mathcal{X})$ with densities p and q respectively. Then, the f-divergence between P and Q is defined as*

$$D_f(P\|Q) = \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}\mu(x)\,,$$

with the convention $f(0) = \lim_{t\to 0^+} f(t)$ and $0f(p/0) = p \lim_{t \to 0^+} t\, f(1/t)$.

Note that the non-negativity condition on f is without loss of generality.[1] Since f is convex and nonnegative with $f(1) = 0$, we have that f is non-increasing on $(0, 1]$ and non-decreasing

[1] The generator $\hat{f}(t) = f(t) + c(t-1)$ yields the same f-divergence as a convex function f with $f(1) = 0$ for all $c \in \mathbb{R}$. By choosing c such that $f'(1) = 0$, we get that $\hat{f}$ is minimized at $t = 1$. This ensures non-negativity: $\inf_{t>0} \hat{f}(t) = \hat{f}(1) = 0$.

on $[1, \infty)$. The conjugate generator to f is the function $f^* : (0, \infty) \to [0, \infty)$ defined by[2]

$$f^*(t) = t f(1/t) ,$$

where again we define $f^*(0) = \lim_{t \to 0^+} f^*(t)$. Since f^* can be constructed by the perspective transform of f, it is also convex. We can verify that $f^*(1) = 0$ and $f^*(t) \ge 0$ for all $t \in (0, \infty)$, so it defines another divergence D_{f^*}. We call it the *conjugate divergence* to D_f since

$$D_{f^*}(P \| Q) = D_f(Q \| P) .$$

The divergence D_f is symmetric if and only if $f = f^*$, and we write it as $D_f(P, Q)$ to emphasize the symmetry.

Example 2. *We give a few examples of f-divergences.*

(a) KL divergence: It is an f-divergence generated by $f_{\mathrm{KL}}(t) = t \log t - t + 1$.

(b) Interpolated KL divergence: For $\lambda \in (0, 1)$, the interpolated KL divergence is given by

$$\mathrm{KL}_\lambda(P \| Q) = \mathrm{KL}(P \| \lambda P + (1 - \lambda) Q) .$$

It is an f-divergence whose generator can be obtained from the upcoming Property 5.

(c) Jensen-Shannon divergence: The Jensen-Shannon Divergence is defined as

$$D_{\mathrm{JS}}(P, Q) = \frac{1}{2}\mathrm{KL}_{1/2}(P \| Q) + \frac{1}{2}\mathrm{KL}_{1/2}(Q \| P).$$

More generally, we have the λ-skew Jensen-Shannon Divergence (Nielsen and Bhatia, 2013), which is defined for $\lambda \in (0, 1)$ as $D_{\mathrm{JS},\lambda} = \lambda \mathrm{KL}_\lambda(P \| Q) + (1 - \lambda)\mathrm{KL}_{1-\lambda}(Q \| P)$. This is an f-divergence generated by

$$f_{\mathrm{JS},\lambda}(t) = \lambda t \log\left(\frac{t}{\lambda t + 1 - \lambda}\right) + (1 - \lambda) \log\left(\frac{1}{\lambda t + 1 - \lambda}\right) .$$

(d) Interpolated χ^2 divergence: Similar to the interpolated KL divergence, we can define the interpolated χ^2 divergence $D_{\chi^2,\lambda}$ and the corresponding generator $f_{\chi^2,\lambda}$ for $\lambda \in (0, 1)$ as

$$D_{\chi^2,\lambda}(P \| Q) = D_{\chi^2}(P \| \lambda P + (1 - \lambda) Q) \quad \text{and} \quad f_{\chi^2,\lambda}(t) = \frac{(t-1)^2}{\lambda t + 1 - \lambda} .$$

The usual χ^2 divergence is obtained in the limit $\lambda \to 0$.

3. Generalizing Divergence Frontiers with f-Divergences

In this section, we start with the notion of KL divergence frontiers from (Djolonga et al., 2020) and define f-divergence frontiers in Section 3.1. We define three scalar summaries of the frontier in Section 3.2 and study their properties in Section 3.3.

[2] The conjugacy between f and f^*, also known as *Csiszár conjugacy*, is unrelated to the Fenchel or Lagrange duality in convex analysis. This notion of conjugacy is related to the perspective transform $g(t, s) = s\, f(t/s)$.

3.1 Tradeoff Curves to Evaluate Generative Models

Consider a generative model $Q \in \mathcal{P}(\mathcal{X})$ which attempts to model the target distribution $P \in \mathcal{P}(\mathcal{X})$. It has been argued in (Sajjadi et al., 2018; Kynkäänniemi et al., 2019) that one must consider two types of costs to evaluate Q with respect to P: (a) a type I cost incurred from generating poor-quality data, which is the mass of Q that has low or zero probability mass under P, and (b) a type II cost incurred from a failure to capture the diversity of the real data, which is the mass of P that Q does not adequately capture.

Suppose P and Q are uniform distributions on their supports, and R is uniform on the union of their supports. Then, the type I cost is the mass of $\mathrm{Supp}(Q) \setminus \mathrm{Supp}(P)$, or equivalently, the mass of $\mathrm{Supp}(R) \setminus \mathrm{Supp}(P)$. We measure this using the surrogate $\mathrm{KL}(Q\|R)$, which is large if there exists an atom $\boldsymbol{x}$ such that $Q(\boldsymbol{x})$ is large but $R(\boldsymbol{x})$ is small. Likewise, the type II cost is measured by $\mathrm{KL}(P\|R)$. When P and Q are not constrained to be uniform, it is not clear what the measure R should be. Djolonga et al. (2020) propose to vary R over all possible probability measures and consider the Pareto frontier of the multi-objective optimization $\min_R \big(\mathrm{KL}(P\|R), \mathrm{KL}(Q\|R)\big)$. This leads to a curve called the *divergence frontier*, illustrated in Figure 1), and is reminiscent of the precision-recall curve in binary classification. See (Pepe, 2000; Cortes and Mohri, 2005; Clémençon and Vayatis, 2009; Clémençon and Vayatis, 2010; Flach, 2012) and references therein on trade-off curves in machine learning.

It was shown in (Djolonga et al., 2020, Props. 1 and 2) that the divergence frontier $\mathcal{F}(P, Q)$ of probability measures P and Q is carved out by mixtures $R_\lambda = \lambda P + (1-\lambda)Q$ for $\lambda \in (0, 1)$. We present an elementary proof for completeness.

Property 3. *Consider two distributions P, Q with finite support. Then, the Pareto frontier for the pair of objectives $\big(\mathrm{KL}(P\|\cdot), \mathrm{KL}(Q\|\cdot)\big)$ is given by*

$$\mathcal{F}(P,Q) = \Big\{ \big(\mathrm{KL}(P\|R_\lambda),\, \mathrm{KL}(Q\|R_\lambda)\big) \,:\, \lambda \in (0,1) \Big\}, \tag{3}$$

where $R_\lambda = \lambda P + (1-\lambda)Q$. In other words, there does not exist any distribution R such that $\mathrm{KL}(P|R) < \mathrm{KL}(P|R_\lambda)$ and $\mathrm{KL}(Q|R) < \mathrm{KL}(Q|R_\lambda)$ simultaneously for any $\lambda \in (0, 1)$.

Proof The convexity of $\mathrm{KL}(P\|\cdot), \mathrm{KL}(Q\|\cdot)$ allows us to compute the Pareto frontier $\mathcal{F}(P,Q)$ exactly by minimizing linear combinations of the objectives. Concretely, we have from (Miettinen, 2012, Thms. 3.4.5 & 3.5.4) that

$$\begin{aligned} \mathcal{F}(P,Q) = \Big\{ \big(\mathrm{KL}(P\|R_\lambda^\star), \mathrm{KL}(P\|R_\lambda^\star)\big) \,:\, \lambda \in [0,1] \Big\}, \quad \text{where} \\ R_\lambda^\star \in \arg\min_R \{ \lambda\, \mathrm{KL}(P\|R) + (1-\lambda)\, \mathrm{KL}(Q\|R) \}\,. \end{aligned}$$

Simple algebra gives us the identity

$$\lambda\, \mathrm{KL}(P\|R) + (1-\lambda)\, \mathrm{KL}(Q\|R) = \lambda\, \mathrm{KL}(P\|R_\lambda) + (1-\lambda)\, \mathrm{KL}(Q\|R_\lambda) + \mathrm{KL}(R_\lambda\|R)\,.$$

The first two terms of the right-hand side are independent of R and the last term is minimized at $R = R_\lambda$. Therefore, $R_\lambda^\star = R_\lambda$. ■

In this work, we consider a more general family of f-divergence frontiers.

Definition 4. *The f-divergence frontier $\mathcal{F}_f(P, Q)$ for two distributions $P, Q \in \mathcal{P}(\mathcal{X})$ and a divergence generator function f satisfying $f(0) < \infty$ and $f^*(0) = \infty$ is defined as*

$$\mathcal{F}_f(P, Q) = \Big\{ \big(D_f(P \| R_\lambda),\, D_f(Q \| R_\lambda) \big) \,:\, \lambda \in (0,1) \Big\} ,$$

where $R_\lambda = \lambda P + (1-\lambda) Q$.

The condition $f(0) < \infty$ ensures that $D_f(P\|R_\lambda)$ and $D_f(Q\|R_\lambda)$ are finite for $0 < \lambda < 1$, so the f-divergence frontier is well defined. The condition $f^*(0) = \infty$ mimics the behavior of the KL divergence so that $D_f(P\|Q) = \infty$ when $P \not\ll Q$ and $D_f(Q\|P) = \infty$ when $Q \not\ll P$. This allows the divergence curve to grow to infinity as λ approaches the endpoints of $(0, 1)$ if the supports of P and Q are not identical. When f is not specified, we refer to the KL divergence frontier defined above—it corresponds to $f(t) = t \log t - t + 1$.

Each coordinate of the f-divergence frontier is itself an f-divergence as we show next.

Property 5. *Consider the f-divergence D_f generated by the convex function f. For any $\lambda \in (0, 1)$, we have that $D_f(P\|\lambda P + (1-\lambda)Q) = D_{f_\lambda}(P\|Q)$ and $D_f(Q\|\lambda P + (1-\lambda)Q) = D_{f_{1-\lambda}}(Q\|P)$, where $f_\lambda : (0, \infty) \to \mathbb{R}_+$ is given by*

$$f_\lambda(t) = (\lambda t + 1 - \lambda)\; f\left(\frac{t}{\lambda t + 1 - \lambda} \right) . \tag{4}$$

Further, D_{f_λ} is a valid f-divergence in that it satisfies the conditions of Definition 1: f_λ is convex, non-negative and $f_\lambda(1) = 0$. Moreover, if f is twice differentiable with $f''(t) > 0$ for all $t > 0$, then f_λ is strictly convex with $f_\lambda''(t) > 0$ for all $t > 0$.

Proof We have $f_\lambda \geq 0$ and $f_\lambda(1) = 0$ by definition. In order to establish the convexity of f_λ, observe that $f_\lambda(t) = (g \circ h_\lambda)(t)$, where $g(t, s) = s\, f(t/s)$ is the perspective transform of f, and $h_\lambda(t) = (t, \lambda t + 1 - \lambda) \in \mathbb{R}_+^2$ is a linear map. The perspective g of a convex function f is convex, and convexity is preserved upon composition with a linear map h_λ, so f_λ is convex. Finally, $D_f(P\|\lambda P + (1-\lambda)Q) = D_{f_\lambda}(P\|Q)$ and $D_f(Q\|\lambda P + (1-\lambda)Q) = D_{f_{1-\lambda}}(Q\|P)$ can be verified from the definition.

To show the strict convexity of f_λ, we calculate

$$f_\lambda''(t) = \frac{(1-\lambda)^2}{(\lambda t + 1 - \lambda)^3}\, f''\left(\frac{t}{\lambda t + 1 - \lambda} \right) > 0$$

under the given assumptions. ■

3.2 Scalar Summaries of Divergence Frontiers

We define three summaries of divergence frontiers.

Area Summary. The first summary is inspired by the area under the curve (e.g. Flach, 2012)—a common strategy to summarize tradeoff curves in machine learning. Divergence frontiers, however, can be unbounded. For instance, as $\lambda \to 1$, we have $\mathrm{KL}(Q\|R_\lambda) \to \mathrm{KL}(Q\|P)$, which can be unbounded. The same holds for f-divergence frontiers because

$f^*(0) = \infty$. Therefore, we define Mauve to be the area under a monotonic transformation of the f-divergence frontier:

$$\textsc{Mauve}_f(P, Q) = \mathrm{AUC}\left(\left\{\left(\exp(-cx), \exp(-cy)\right) \,:\, (x, y) \in \mathcal{F}_f(P, Q)\right\} \cup \{(1,0), (0,1)\}\right) . \tag{5}$$

Here, $c > 0$ is a scaling constant that changes the numerical value of Mauve, but not its induced ordering over multiple models $Q_1, \dots, Q_n$. $\textsc{Mauve}_f(P, Q)$ is always bounded between 0 and 1 with larger values denoting a greater similarity between P and Q.

Integral Summary. For the second summary of the divergence frontier, we take inspiration from the minimax theory of hypothesis testing, where the goal is also to study two types of errors and it is common to theoretically analyze their linear combination; see, e.g., (Ingster and Suslina, 2003, Sec. 1.2) and (Cai et al., 2011, Thm. 7). Similarly, we consider a linear combination of the two costs that are the two coordinates of the divergence frontier:

$$L_{f,\lambda}(P, Q) := \lambda\, D_f(P\|R_\lambda) + (1-\lambda) D_f(Q\|R_\lambda) . \tag{6}$$

Note that, for the KL divergence, R_λ is exactly the minimizer of the linearized objective $\lambda\,\mathrm{KL}(P\|R) + (1-\lambda)\mathrm{KL}(Q\|R)$ according to Property 3. In this case, L_λ is also known as the λ-skew Jensen-Shannon Divergence (cf. Example 2).

The linearized cost $L_{f,\lambda}$ depends on the choice of λ. To remove this dependency, we define an integral summary as

$$\mathrm{FI}_f(P, Q) := 2\int_0^1 L_{f,\lambda}(P, Q)\,\mathrm{d}\lambda . \tag{7}$$

We can interpret the frontier integral as the average linearized cost over $\lambda \in (0, 1)$. The constant of 2 is arbitrary and is chosen so that $\mathrm{FI}_{\mathrm{KL}}$ is bounded above by 1, as we shall momentarily see in Section 3.3.

Mid-point Summary. The third summary is a generalization of the Jensen-Shannon divergence, defined to be the linearized cost with weight $\lambda = 1/2$, i.e.,

$$\mathrm{Mid}_f(P, Q) := L_{f,1/2}(P, Q) = \frac{1}{2}\, D_f(P\|R_{1/2}) + \frac{1}{2} D_f(Q\|R_{1/2}) . \tag{8}$$

When f is the generator of the KL (resp. χ^2) divergence, it recovers the Jensen-Shannon (resp. Le Cam) divergence. This summary is intuitively close to the area summary as illustrated in Figure 3.

3.3 Properties of Divergence Frontier Summaries

We study some properties of the area summary Mauve.

Property 6. *Fix an f-divergence $D_f(\cdot\|\cdot)$ such that $f(0) < \infty$ and a scaling constant $c > 0$. For any two distributions P, Q with finite support, the area summary* $\textsc{Mauve}(P, Q)$ *satisfies the following:*

(a) $0 \le \textsc{Mauve}_f(P, Q) = \textsc{Mauve}_f(Q, P) \le 1$,

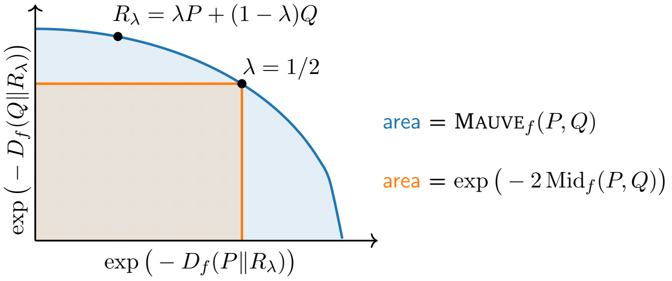

$R_\lambda = \lambda P + (1-\lambda) Q$
$\lambda = 1/2$
area = $\text{MAUVE}_f(P, Q)$
area = $\exp\big(-2\,\text{Mid}_f(P,Q)\big)$
$\exp\big(-D_f(Q\|R_\lambda)\big)$
$\exp\big(-D_f(P\|R_\lambda)\big)$

Figure 3: Relationship between the area summary MAUVE_f and the mid-point summary Mid_f. MAUVE is the area under the blue curve, while the mid-point summary Mid is related to the area under the orange rectangle.

(b) $\text{MAUVE}_f(P,P) = 1$, *and*

(c) if f is strictly convex, $\text{MAUVE}_f(P,Q) = 1$ *if and only if* $P = Q$.

Proof The curve $(\exp(-cx), \exp(-cy))$ for $(x,y) \in \mathcal{F}_f$ always lies within the unit square, so $0 \le \text{MAUVE}_f(P,Q) \le 1$. If $P = Q$, then $D_f(P\|R_\lambda) = D_f(Q\|R_\lambda) = 0$ for all $\lambda \in (0,1)$, so that $\text{MAUVE}_f(P,Q)$ is simply the area of the unit square. Conversely, if $P \ne Q$, we have that $D_f(P\|R_\lambda) \ne 0$ and $D_f(Q\|R_\lambda) \ne 0$ for any $\lambda \in (0,1)$ whenever f is strictly convex. Therefore, the curve $(\exp(-cx), \exp(-cy))$ for $(x,y) \in \mathcal{F}_f$ lies strictly within the unit square and $\text{MAUVE}_f(P,Q) < 1$. ■

We now turn to the integral summary.

Property 7. *The integral summary* FI *of the f-divergence frontier defined by a convex generator f satisfies the following properties:*

(a) FI_f *is an f-divergence generated by the convex function*

$$\tilde{f}(t) = 2\int_0^1 \Big(\lambda\, f_\lambda(t) + (1-\lambda) f^*_{1-\lambda}(t)\Big) \mathrm{d}\lambda\,,$$

where f_λ is as defined in (4).

(b) $\text{FI}_f(P,Q) = \text{FI}_f(Q,P)$.

(c) $0 \le \text{FI}_f(P,Q) \le 4\int_0^1 \lambda f^*(\lambda)\mathrm{d}\lambda + \frac{2}{3} f(0)$.

(d) If f is twice differentiable with $f''(t) > 0$ for all $t > 0$, we have $\text{FI}_f(P,Q) = 0$ *if and only if* $P = Q$.

Proof We denote $\bar\lambda = 1 - \lambda$. For the first part, we have from Property 5,

$$\text{FI}_f(P,Q) = 2\int_0^1 \big(\lambda D_{f_\lambda}(P\|Q) + \bar\lambda D_{(f_{\bar\lambda})^*}(P\|Q)\big)\,\mathrm{d}\lambda = D_{\tilde f}(P\|Q)\,,$$

by using the definition of f-divergences. Note that $\tilde f$ is a convex function as it is the positive linear combination of a family of convex functions. We also directly verify that

$\tilde{f}(t) \ge \tilde{f}(1) = 0$ for all $t > 0$, so $D_{\tilde{f}}$ is a well-defined f-divergence. For the second part, we get

$$(\tilde{f})^*(t) = t\tilde{f}(1/t) = 2\int_0^1 \Big(\lambda\, f_\lambda^*(t) + (1-\lambda) f_{1-\lambda}(t)\Big) \mathrm{d}\lambda \ = \tilde{f}(t)\,,$$

where the last equality follows by substituting $\lambda' = 1-\lambda$. Therefore, $\mathrm{FI}_f(Q,P) = D_{\tilde{f}}(Q\|P) = D_{\tilde{f}^*}(P\|Q) = D_{\tilde{f}}(P\|Q) = \mathrm{FI}_f(P,Q)$. For the third part, we use the upper bound on $L_{f,\lambda}$ from Proposition 19 in Appendix A to get

$$\mathrm{FI}_f(P,Q) = 2\int_0^1 L_{f,\lambda}(P\|Q)\,\mathrm{d}\lambda \le 2\int_0^1 \big(\lambda f^*(\lambda) + \bar\lambda f^*(\bar\lambda) + 2\lambda\bar\lambda f(0)\big)\,\mathrm{d}\lambda\,.$$

Simplifying this integral gives the third part. For the final part, we note that $f_\lambda''(t) > 0$ and $(f_{\bar\lambda}^*)''(t) > 0$ for all $t > 0$ from Property 5. This gives

$$(\tilde{f})''(t) = 2\int_0^1 \Big(\lambda\, f_\lambda''(t) + (1-\lambda)(f_{1-\lambda}^*)''(t)\Big) \mathrm{d}\lambda \ > 0\,.$$

This implies that $\tilde{f}$ is strictly convex. Therefore, $D_{\tilde{f}}(P\|Q) = 0$ iff $P = Q$. ■

We can instantiate this property for common divergences. The integral summary $\mathrm{FI}_{\mathrm{KL}}$ of the KL divergence frontier is generated by

$$\tilde{f}_{\mathrm{KL}}(t) = \frac{t+1}{2} - \frac{t}{t-1}\log t\,,$$

with the understanding that $\tilde{f}_{\mathrm{KL}}(1) = \lim_{t\to 1} \tilde{f}_{\mathrm{KL}}(t) = 0$. Similarly, the corresponding expression for the integral summary of the χ^2 divergence frontier is

$$\tilde{f}_{\chi^2}(t) = \frac{t^2+t+1}{t-1}\,\log t - \frac{3}{2}\,(t+1)\ .$$

We have that $\mathrm{FI}_{\mathrm{KL}}$ and FI_{χ^2} are upper bounded by 1 and 2 respectively.

Lastly, we turn to the mid-point summary.

Property 8. *The mid-point summary* Mid_f *of the* f*-divergence frontier defined by a generator* f *satisfies the following properties:*

(a) Mid_f *is an* f*-divergence generated by the convex function* $f_{1/2}$ *as defined in* (4).

(b) $\mathrm{Mid}_f(P,Q) = \mathrm{Mid}_f(Q,P)$.

(c) $0 \le \mathrm{Mid}_f(P,Q) \le \frac{1}{2}\left(f(0) + f(2)\right)$.

(d) If f *is twice differentiable with* $f''(t) > 0$ *for all* $t > 0$*, we have* $\mathrm{Mid}_f(P,Q) = 0$ *if and only if* $P = Q$.

Proof The first, second, and fourth parts follow directly from Property 5. The third part is a consequence of Proposition 19 in Appendix A. ■

4. Practical Computation of the Divergence Frontier and its Summaries

In this section, we consider how to compute Mauve and related divergence frontier summaries for high dimensional distributions of text or images. We usually do not have access to the target distribution P representing human-written text or real-world images. While the model likelihood $Q(\boldsymbol{x})$ can be evaluated for some generative model Q such as language models for text, it might not be available for others such as generative adversarial networks for images. Therefore, we only assume access to the distributions P and Q via i.i.d. samples.

Given two independent samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} P$ and $\boldsymbol{x}'_1, \dots, \boldsymbol{x}'_m \overset{\text{i.i.d.}}{\sim} Q$, we wish to estimate the summaries $\text{Mauve}_f(P, Q)$, $\text{FI}_f(P, Q)$, or $\text{Mid}_f(P, Q)$ using these samples. We will often assume equal sample sizes $m = n$ for simplicity, especially when stating bounds. In real image or text applications, the distributions P and Q are typically discrete distributions whose support size is too large to enumerate. For instance, neural language models induce a probability distribution over documents of text. Thus, we cannot tractably compute the f-divergences required by the divergence frontiers or their scalar summaries in closed form. Instead, we consider four different estimation methods:

- **Vector Quantization**: We quantize the empirical distributions $\hat{P}_n = (1/n)\sum_{i=1}^n \delta_{\boldsymbol{x}_i}$ and $\hat{Q}_m = (1/m)\sum_{j=1}^m \delta_{\boldsymbol{x}'_j}$ into k-dimensional multinomial distributions $\hat{P}_{n,k}$ and $\hat{Q}_{m,k}$, where k is a hyperparameter. We then estimate the divergence frontier by the plug-in estimator $\mathcal{F}_f(\hat{P}_{n,k}, \hat{Q}_{m,k})$, from which the corresponding summaries Mauve, FI, and Mid can be estimated. This approach can also be used with add-constant distribution estimators in place of empirical distributions; see Table 2 for some examples.

- **Nearest-neighbor estimation**: We endow the space $\mathcal{X}$ with a metric $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ and consider the set $N_k(\boldsymbol{x})$ of the k-nearest neighbor of $\boldsymbol{x}$ from the union of $X = \{\boldsymbol{x}_i\}_{i=1}^n$ and $X' = \{\boldsymbol{x}'_j\}_{j=1}^m$. We estimate the likelihood ratio $P(\boldsymbol{x}'_j)/Q(\boldsymbol{x}'_j)$ based on the ratio $|N_k(\boldsymbol{x}'_j) \cap X| / |N_k(\boldsymbol{x}'_j) \cap X'|$ for $j = 1, \dots, m$. This likelihood ratio can then be used to estimate the required f-divergences.

- **Classifier-based estimation**: We train a classifier over samples $\{(\boldsymbol{x}_1, +1)\}_{i=1}^{n'} \cup \{(\boldsymbol{x}'_j, -1)\}_{j=1}^{m'}$ and use this to estimate the likelihood ratio $P(\boldsymbol{x})/Q(\boldsymbol{x})$ over the remaining $n - n' + m - m'$ samples. This likelihood ratio can then be used to estimate the required f-divergences.

- **Parametric approximation**: Given an embedding $\varphi : \mathcal{X} \to \mathbb{R}^d$, we make a parametric assumption that the pushforward distributions $\varphi_\sharp P = \mathcal{N}(\mu_P, \Sigma_P)$ and $\varphi_\sharp Q = \mathcal{N}(\mu_Q, \Sigma_Q)$ with unknown parameters $\mu_P, \Sigma_P, \mu_Q, \Sigma_Q$. We estimate $\hat{\mu}_P, \hat{\Sigma}_P, \hat{\mu}_Q, \hat{\Sigma}_Q$ from data and use $\mathcal{F}_f\big(\mathcal{N}(\hat{\mu}_P, \hat{\Sigma}_P), \mathcal{N}(\hat{\mu}_Q, \hat{\Sigma}_Q)\big)$ as an estimate that is computed by numerical integration. Although this approach is widely used in practice, it has no theoretical guarantees. Therefore, we defer its discussion to Appendix C and compare its empirical performance with other methods in Section 7.3.

In the rest of this section, we consider each in detail. In full generality, we will focus on estimating f-divergences from samples. The results on estimating the f-divergence frontier $\mathcal{F}_f(P, Q)$ follow as corollaries because each point on the frontier is itself an f-divergence (Property 5).

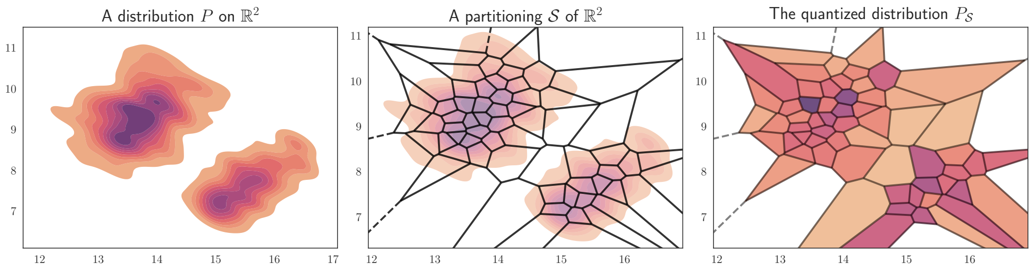


Figure 4: Illustration of the quantization $P_{\mathcal{S}}$ of a distribution P over the Euclidean plane $\mathbb{R}^2$ under a partition $\mathcal{S}$.

4.1 Estimation via Vector Quantization

Given a k-partition $\mathcal{S} = \{S_1, \dots, S_k\}$ of the space $\mathcal{X}$, we define the quantization of P over $\mathcal{S}$ as $P_{\mathcal{S}} = \big(P(S_1), \dots, P(S_k)\big)$. Then, $P_{\mathcal{S}}$ and $Q_{\mathcal{S}}$ are multinomial distributions over k atoms; they are piecewise constant approximations of P and Q similar to histograms as illustrated in Figure 4. The quantization approach to estimating the divergence frontier consists of two approximations:

- approximating the intractable divergence frontier $\mathcal{F}_f(P, Q)$ with the lower-dimensional counterpart $\mathcal{F}_f(P_{\mathcal{S}}, Q_{\mathcal{S}})$, and
- estimating this frontier $\mathcal{F}_f(P_{\mathcal{S}}, Q_{\mathcal{S}})$ with its plug-in estimator $\mathcal{F}_f(\hat{P}_{\mathcal{S},n}, \hat{Q}_{\mathcal{S},m})$, where $\hat{P}_{\mathcal{S},n} = \big(n^{-1} \sum_{i=1}^n \mathbb{1}\{\boldsymbol{x}_i \in S_l\}\big)_{l=1}^k$ is the empirical distribution of $P_{\mathcal{S}}$, and $\hat{Q}_{\mathcal{S},m}$ is the corresponding empirical distribution of $Q_{\mathcal{S}}$

In practice, the best quantization schemes are data-dependent, such as k-means clustering or lattice-type vector quantization of dense representations of images or text; we will discuss this in more detail in Section 4.1.2.

When the two distributions P and Q have long tails, the empirical estimators $\hat{P}_{\mathcal{S},n}$ and $\hat{Q}_{\mathcal{S},m}$ can be of poor quality due to the *missing mass* phenomenon (Good, 1953), i.e., some probability masses do not appear in the finite sample. This is illustrated in Figure 5. A widely used technique to address such a challenge is the *add-constant* smoothing (see, e.g., Krichevsky and Trofimov, 1981). This approach adds a small constant b to the counts of each bin and normalizes these pseudo-counts to form a normalized probability distribution. Precisely, the add-b estimator of $P_{\mathcal{S}}$ is defined as

$$\hat{P}^b_{\mathcal{S},n} = \left(\frac{b + \sum_{i=1}^n \mathbb{1}\{\boldsymbol{x}_i \in S_l\}}{n + kb}\right)_{l=1}^k . \tag{9}$$

Other estimators suitable for this regime have also been considered in the literature such as the Good-Turing estimator (Orlitsky and Suresh, 2015) and absolute discounting (Falahatgar et al., 2017).

4.1.1 Estimation Error Bounds

The total estimation error of the divergence frontier consists of two parts: (a) the statistical error in estimating $\mathcal{F}_f(P_{\mathcal{S}}, Q_{\mathcal{S}})$ from samples, and (b) the quantization error in passing from P, Q to $P_{\mathcal{S}}, Q_{\mathcal{S}}$. For simplicity, we assume in this subsection that $m = n$. In what

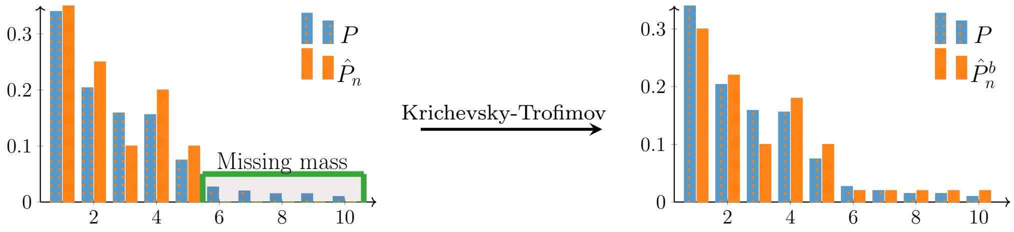


Figure 5: Left: Missing mass of a sample corresponds to those entries $l \in \mathrm{Supp}(P)$ that do not appear in the sample, i.e., $\hat{P}_{n,l} = 0$. **Right**: Add-constant smoothing adds a constant b to counts of each bin $l \in \mathrm{Supp}(P)$, including those that do not appear in the sample. Krichevsky–Trofimov smoothing corresponds to $b = 1/2$.

follows, we establish a statistical error bound of order $O(\sqrt{k/n})$ and show that there exists a quantization scheme with error $O(1/k)$. The theory suggests that we can balance the two errors at $k = \Theta\big(n^{1/3}\big)$.

Statistical Estimation Error. We establish a statistical bound on estimating a general f-divergence $D_f(P\|Q)$ between discrete distributions P, Q using their plug-in estimators $\hat{P}_n, \hat{Q}_n$ from samples, respectively. To this end, we require the generator f and its conjugate f^* to satisfy some smoothness and tail assumptions.

Assumption 9. *The generator f is twice continuously differentiable with $f'(1) = 0$. Furthermore,*

(A1) *We have $C_0 := f(0) < \infty$ and $C_0^* := f^*(0) < \infty$.*

(A2) *There exist constants $C_1, C_1^* < \infty$ such that for every $t \in (0, 1)$, we have,*

$$|f'(t)| \le C_1 \left(1 \vee \log(1/t)\right), \quad \textit{and}, \quad |(f^*)'(t)| \le C_1^* \left(1 \vee \log(1/t)\right) .$$

(A3) *There exist constants $C_2, C_2^* < \infty$ such that for every $t \in (0, \infty)$, we have,*

$$\frac{t}{2} f''(t) \le C_2 \,, \quad \textit{and}, \quad \frac{t}{2}(f^*)''(t) \le C_2^* \,.$$

Some boundedness assumption is necessary since the minimax quadratic risk of estimating the KL divergence over all discrete distributions with k atoms is always infinity (Bu et al., 2018). Assumption **(A1)** is a necessary and sufficient condition for $D_f(P\|Q)$ and $D_{f^*}(P\|Q)$ to remain bounded for all distributions P, Q. Assumption **(A2)** guarantees that f is approximately Lipschitz and cannot vary too fast, while **(A3)** is a technical assumption that helps control the variation of f around zero.

These assumptions hold for many f-divergences, as shown in Table 1. Notably, they hold for the $\mathrm{FI}_{\mathrm{KL}}$ and $\mathrm{Mid}_{\mathrm{KL}}$, as well as the coordinates of the KL and χ^2 divergence frontiers.

We now turn to the statistical error bound. When both P and Q are supported on a finite alphabet with k items, a natural strategy is to exploit the smoothness properties of the f-divergence, namely Assumption **(A2)**. This gives a naïve upper bound $O(L\sqrt{k/n})$ on the absolute error, where $L = C_1 \log(1/p_*)$ with $p_* = \min_{l \in \mathrm{Supp}(P)} P_l$ reflects the smoothness of the f-divergence. The dependency on p_* requires P to have finite support and a short tail. However, in many real-world applications, the distributions can either be supported

f-divergence	Satisfies Assumptions?	C_0	C_0^*	C_1	C_1^*	C_2	C_2^*
KL	No	1	∞				
Interpolated KL	Yes	$\bar\lambda$	$\log\frac{1}{\lambda} - \bar\lambda$	1	$\frac{\bar\lambda^2}{\lambda}$	$\frac{1}{2}$	$\frac{\bar\lambda}{8\lambda}$
Jensen-Shannon (JS) / $\mathrm{Mid}_{\mathrm{KL}}$	Yes	$\frac{1}{2}\log 2$	$\frac{1}{2}\log 2$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$
Skew JS	Yes	$\bar\lambda \log\frac{1}{\bar\lambda}$	$\lambda\log\frac{1}{\lambda}$	λ	$\bar\lambda$	$\frac{\lambda}{2}$	$\frac{\bar\lambda}{2}$
$\mathrm{FI}_{\mathrm{KL}}$	Yes	$\frac{1}{2}$	$\frac{1}{2}$	4	4	$\frac{1}{2}$	$\frac{1}{2}$
Interpolated χ^2	Yes	$\frac{1}{\bar\lambda}$	$\frac{1}{\lambda}$	$\frac{2}{\bar\lambda^2}$	$\frac{2}{\lambda^2}$	$\frac{4}{27\lambda\bar\lambda^2}$	$\frac{4}{27\lambda^2\bar\lambda}$
Le Cam / Mid_{χ^2}	Yes	$\frac{1}{2}$	$\frac{1}{2}$	2	2	$\frac{8}{27}$	$\frac{8}{27}$
Squared Hellinger	No	1	1	∞	∞		

Table 1: Examples of f-divergences and whether they satisfy Assumptions **(A1)**-**(A3)**. Here, $\lambda \in (0,1)$ is a parameter of the interpolated or skew divergences, and we define $\bar\lambda := 1-\lambda$.

on a countable set or have long tails (Chen and Goodman, 1999; Wang et al., 2017). By considering the *missing mass* in the sample, that is the total probability mass that does not appear in the finite sample (Good, 1953), we can obtain a bound that is independent of p_*. We refer to Figure 5 (left) for an illustration of the missing mass.

Theorem 10. *Assume that* $k := |\mathrm{Supp}(P)| \vee |\mathrm{Supp}(Q)| \in \mathbb{N} \cup \{\infty\}$. *Let* $n \ge 3$, $c_1 := C_1 + C_1^*$, *and* $c_2 := C_2 \vee C_0^* + C_2^* \vee C_0$. *Under Assumption 9, we have,*

$$\begin{aligned}\mathbb{E}|D_f(P\|Q) - D_f(\hat P_n\|\hat Q_n)| \le \big(C_1 \log n + C_0^* \vee C_2\big)\alpha_n(P) + \big(C_1^* \log n + C_0 \vee C_2^*\big)\alpha_n(Q) \\ + \big(C_1 + C_0^* \vee C_2\big)\beta_n(P) + \big(C_1^* + C_0 \vee C_2^*\big)\beta_n(Q)\,,\end{aligned} \tag{10}$$

where $\alpha_n(P) = \sum_{l=1}^k \sqrt{n^{-1}P_l}$ *and* $\beta_n(P) = \mathbb{E}\big[\sum_{l:\hat P_n(l)=0} P_l \max\{1, \log(1/P_l)\}\big]$. *Furthermore, if* $k < \infty$, *then*

$$\mathbb{E}|D_f(P\|Q) - D_f(\hat P_n\|\hat Q_n)| \le \big(c_1 \log n + c_2\big)\left(\sqrt{\frac{k}{n}} + \frac{k}{n}\right). \tag{11}$$

In particular, for the Frontier Integral, it gives a statistical error bound of

$$\mathbb{E}|\mathrm{FI}(\hat P_n, \hat Q_n) - \mathrm{FI}(P,Q)| \le C\left(\sqrt{\frac{k}{n}} + \frac{k}{n}\right)\log n\,, \tag{12}$$

where C *is some absolute constant.*

Some remarks about the bounds in Theorem 10 are as follows. First, the bound (10) holds for any distributions with a countable support. Second, it does not depend on p_* and is adapted to the tail behavior of P and Q. For instance, if P is defined as $P_l \propto l^{-2}$ for $l \in [k]$, then $\alpha_n(P) \propto (\log k)/\sqrt{n}$, which is much smaller than $\sqrt{k/n}$ in (11) in terms of the dependency on k. This result justifies the practice of using a large quantization size k

on real data. Third, it captures a parametric rate of convergence, i.e., $O(n^{-1/2})$, up to a logarithmic factor. This rate is not improvable in a related problem of estimating $\mathrm{KL}(P\|Q)$, even with the assumption that P/Q is bounded (Bu et al., 2018). The bound in (11) is a distribution-free bound, assuming k is finite. Note that it also gives an upper bound on the sample complexity by setting the right-hand side of (11) to be ε and solving for n; this is roughly k/ε^2, ignoring constants and log factors.

Proof [Proof Sketch of Theorem 10] We sketch the proof for the $\mathrm{FI}_{\mathrm{KL}}(P,Q) = D_{\tilde f}(P\|Q)$ with full details given in Appendix B.1. The proof relies on a careful analysis of the derivatives of the f-divergence while accounting for the missing mass. We start by defining the bivariate scalar function $\psi(p,q) = q\,\tilde f(p/q)$ where $\tilde f$ is the generator of FI. Then, we have $\mathrm{FI}(P,Q) = \sum_{l=1}^k \psi(P_l, Q_l)$. By the triangle inequality, we have,

$$\left|\mathrm{FI}(\hat P_n, \hat Q_n) - \mathrm{FI}(P,Q)\right| \le \sum_{l=1}^k \underbrace{\left|\psi(\hat P_{n,l}, \hat Q_{n,l}) - \psi(P_l, \hat Q_{n,l})\right|}_{=:\Delta_l} + \underbrace{\left|\psi(P_l, \hat Q_{n,l}) - \psi(P_l, Q_l)\right|}_{=:\Delta_l'} .$$

We bound Δ_l in terms of $|\hat P_{n,l} - P_l|$ so that summing over all coordinates gives a bound on the total variation distance $\|\hat P_n - P\|_{\mathrm{TV}} = \sum_{l=1}^k |\hat P_{n,l} - P_l|$. A first-order Taylor expansion gives the bound

$$\Delta_l \le \sup_{s\in[0,1]} |\psi_p(sP_l + (1-s)\hat P_{n,l}, Q_l)|\ |P_l - \hat P_{n,l}|\,,$$

where ψ_p denotes the partial derivative of ψ w.r.t. its first argument. Unfortunately, as $p \to 0$ for fixed $q \ne 0$, we have that $|\psi_p(p,q)| = |\tilde f'(p/q)| \le \log(q/p) \to \infty$ by Assumption **(A2)**.

We use a two-pronged approach to overcome this issue. First, we take a second-order Taylor expansion and carefully bound the remainder term using Assumption **(A3)** to get

$$\Delta_l \le \frac{1}{2}\,|\hat P_{n,l} - P_l|\ \log\left(\frac{1}{\max\{P_l, \hat P_{n,l}\}}\right) . \tag{13}$$

Secondly, because $\hat P_n$ is an empirical distribution, we only have two possibilities: $\hat P_{n,l} \ge 1/n$ or $\hat P_{n,l} = 0$. The first case gives an additional $\log n$ dependence on the total variation distance (based on Assumption **(A2)**), while the second case is in the missing mass regime. Based on results from the missing mass literature (Berend and Kontorovich, 2012; Mcallester and Ortiz, 2003), we show

$$\beta_n(P) = \mathbb{E}\left[\sum_{l=1}^k \mathbb{I}(\hat P_{n,l} = 0)\ P_l \log\frac{1}{P_l}\right] \le \frac{k\log n}{n}\,,$$

where $\beta_n(P)$ is constructed from the upper bound (13) with $\hat P_{n,l} = 0$. Finally, we bound the total variation term by repeatedly applying Jensen's inequality as

$$\mathbb{E}\|\hat P_n - P\|_{\mathrm{TV}} \le \sum_{l=1}^k \sqrt{\mathbb{E}(\hat P_{n,l} - P_l)^2} = \sum_{l=1}^k \sqrt{\frac{P_l(1-P_l)}{n}} \le \alpha_n(P) \le \sqrt{\frac{k}{n}}\,.$$

■

Braess-Sauer	Krichevsky-Trofimov	Laplace
$b_l = 1/2$ if l does not appear	$b \equiv 1/2$	$b \equiv 1$
$b_l = 1$ if l appears once		
$b_l = 3/4$ if l appears more than once		

Table 2: Add-constant smoothed distribution estimators.

Following Property 5, we can specialize Theorem 10 to show the consistent estimation of the entire f-divergence frontier $\mathcal{F}(P,Q)$.

Proposition 11. *Take an arbitrary $\lambda_0 \in (0,1)$. Suppose we are given distributions P, Q with $k := |\mathrm{Supp}(P)| \vee |\mathrm{Supp}(Q)| \in \mathbb{N} \cup \{\infty\}$. Assume that Assumption 9 holds true for f_λ with $\lambda \in [\lambda_0, 1-\lambda_0]$. If the sample size $n \geq 3$, the bounds in (10) and (11) hold for*

$$\mathbb{E}\left[\sup_{\lambda \in [\lambda_0, 1-\lambda_0]} \left\{ \left| D_f(\hat{P}_n \| \hat{R}_\lambda) - D_f(P \| R_\lambda) \right| + \left| D_f(\hat{Q}_n \| \hat{R}_\lambda) - D_f(Q \| R_\lambda) \right| \right\}\right], \tag{14}$$

where $\hat{R}_\lambda := \lambda \hat{P}_n + (1-\lambda)\hat{Q}_n$, with constants replaced by C/λ_0 for some absolute constant C. In particular, if $\lambda_0 = \lambda_n$ is chosen as $\lambda_n = o(1)$ and $\lambda_n = \omega(\sqrt{k/n}\log n)$, then the expected worst-case error (14) converges to zero at rate $O(\lambda_n^{-1}\sqrt{k/n}\log n)$.

When f is the generator to the KL divergence, Assumption 9 holds for f_λ. Hence, Proposition 11 holds for the KL divergence frontier. In the absence of additional assumptions, the truncation in Proposition 11 is necessary to ensure boundedness of the estimated quantities, since $\mathrm{KL}(P\|R_\lambda)$ is close to $\mathrm{KL}(P\|Q)$ for small λ, and this can be unbounded.

Estimation Error With Smoothing. We bound the statistical error in estimating the divergence $D_f(P\|Q)$ between P and Q using their add-constant estimators $\hat{P}_n^b$ and $\hat{Q}_n^b$ introduced in (9) and illustrated in Figure 5. Again, this result also holds for the $\mathrm{FI}_{\mathrm{KL}}$ and $\mathrm{Mid}_{\mathrm{KL}}$, as well as the coordinates of the KL and χ^2 divergence frontiers. This result is proved in Appendix B.2.

Theorem 12. *Assume that $k := |\mathrm{Supp}(P)| \vee |\mathrm{Supp}(Q)| \in \mathbb{N} \cup \{\infty\}$. Let $n \geq 3$, $c_1 := C_1 + C_1^*$, and $c_2 := C_2 \vee C_0^* + C_2^* \vee C_0$. Under Assumption 9, we have,*

$$\begin{aligned}\mathbb{E}|D_f(P\|Q) - D_f(\hat{P}_n^b \| \hat{Q}_n^b)| \leq & \left(\frac{n\alpha_n(P)}{n+kb} + \gamma_{n,k}(P)\right)\left(C_1 \log\left(\frac{n}{b}+k\right) + C_0^* \vee C_2\right) \\ & + \left(\frac{n\alpha_n(Q)}{n+kb} + \gamma_{n,k}(P)\right)\left(C_1^* \log\left(\frac{n}{b}+k\right) + C_0 \vee C_2^*\right).\end{aligned} \tag{15}$$

where $\gamma_{n,k}(P) = (n+bk)^{-1} bk \sum_{l=1}^{k} |P_l - k^{-1}|$. Furthermore, if $k < \infty$, then

$$\mathbb{E}|D_f(P\|Q) - D_f(\hat{P}_n^b \| \hat{Q}_n^b)| \leq \left(c_1 \log\left(\frac{n}{b}+k\right) + c_2\right) \frac{\sqrt{kn} + 2b(k-1)}{n+kb}. \tag{16}$$

In particular, for the Frontier Integral, it gives a statistical error bound of

$$\mathbb{E}|\mathrm{FI}(\hat{P}_n^b, \hat{Q}_n^b) - \mathrm{FI}(P,Q)| \leq C \frac{\sqrt{kn} + 2b(k-1)}{n+kb} \log\left(\frac{n}{b}+k\right), \tag{17}$$

where C is some absolute constant.

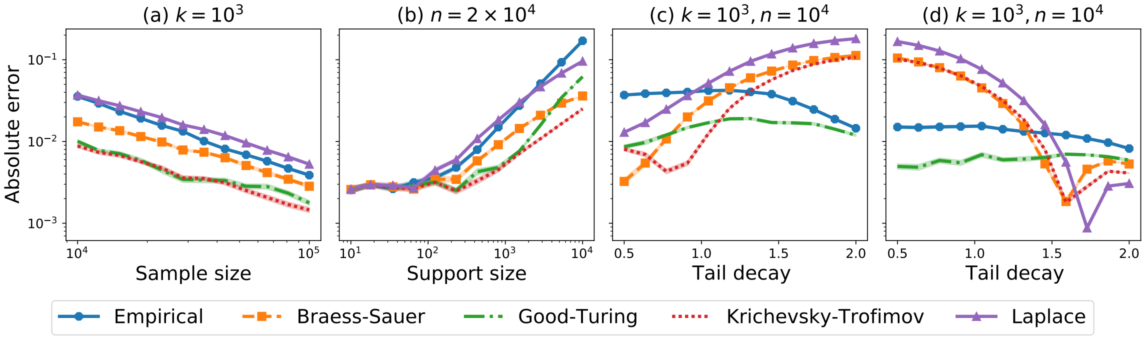


Figure 6: Statistical error with smoothed distribution estimators on synthetic data. **(a)**: Zipf(0) and Dir($\mathbf{1}/2$) with $k = 10^3$; **(b)**: Zipf(0) and Dir($\mathbf{1}/2$) with $n = 2 \times 10^4$; **(c)**: Dir($\mathbf{1}$) and Zipf(r) with $k = 10^3$ and $n = 10^4$; **(d)**: Zipf(2) and Zipf(r) with $k = 10^3$ and $n = 10^4$.

Let us compare the bounds in Theorem 12 with the ones in Theorem 10. For the distribution-dependent bound, the term $\alpha_n(P) \log n$ in (10) is improved by a factor $n/(n+bk)$ in (15). The missing mass term $\beta_n(P)$ is replaced by the total variation distance between P and the uniform distribution on $[k]$ with a factor $bk/(n + bk)$. The improvements in both two terms are most significant when k/n is large. As for the distribution-free bound, when k/n is small, the bound in (16) scales the same as the one in (11); when k/n is large (i.e., bounded away from 0 or diverging), it scales as $O(\log n + \log (k/n) + k^{-1})$ while the one in (11) scales as $O(k \log n/n + k^{-1})$.

Simulations of Smoothing. We conduct a simple simulation study to empirically verify the effectiveness of smoothing. Following the experimental settings used by Orlitsky and Suresh (2015), we consider two types of distributions: (a) the Zipf(r) distribution with $r \in [0, 2]$ where $P_l \propto l^{-r}$. (b) the Dirichlet distribution Dir(α) with $\alpha \in \{\mathbf{1}/2, \mathbf{1}\}$. For each pair (P, Q), we generate i.i.d. samples of size n from each of them and estimate the Frontier Integral from these samples. We compare 4 different smoothed distribution estimators with the empirical distribution ("Empirical") as discussed in (Orlitsky and Suresh, 2015). For each $l \in \mathcal{X}$, let n_l be the number of times l appears in the sample and let φ_t be the number of symbols appearing t times in the sample. The *(modified) Good-Turing* estimator is defined as $\hat{P}^{\mathrm{GT}}_{n,l} \propto n_l$ if $n_l > \varphi_{n_l+1}$ and $\hat{P}^{\mathrm{GT}}_{n,l} \propto \big(\varphi_{n_l+1} + 1\big)(n_l + 1)/\varphi_{n_l}$ otherwise. The remaining three estimators are all based on the add-b smoothing. For the *Braess-Sauer* estimator, the pseudo-count parameter $b = b_l$ is data-dependent and chosen as $b_l = 1/2$ if $n_l = 0$, $b_l = 1$ if $n_l = 1$ and $b_l = 3/4$ otherwise. For the *Krichevsky-Trofimov* estimator, the parameter $b \equiv 1/2$. For the *Laplace* estimator, the parameter $b \equiv 1$. See Table 2 for a summary.

As shown in Figure 6, the smoothed distribution estimators reduce the absolute error. For parts (a) and (b), the Good-Turing and the Krichevsky-Trofimov estimators have the best absolute error. For parts (c) and (d), the Good-Turing estimator is adapted to various regimes of tail-decay, outperforming the empirical estimator. The Krichevsky-Trofimov and Braess-Sauer estimators, on the other hand, exhibit small absolute errors for particular decay regimes. While the smoothed estimators offer a marked improvement when k/n is large (that is, close to 1), the best estimator is problem-dependent. As a rule of thumb, we

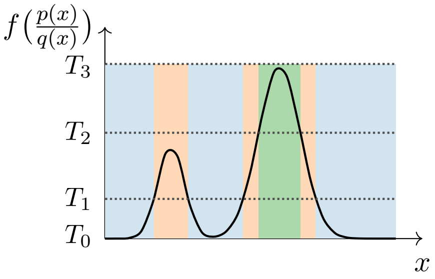


Figure 7: Oracle quantization $\mathcal{S}$ in the estimation of the f-divergence $D_f(P\|Q)$ with $D_f(P_{\mathcal{S}}\|Q_{\mathcal{S}})$, where P and Q have densities p and q. This example shows quantization into $|\mathcal{S}| = 3$ bins: blue, orange, and green. Bin i is given by the set $\{x : f(p(x)/q(x)) \in [T_{i-1}, T_i)\}$.

suggest the Krichevsky-Trofimov estimator which works well in the large k/n regime but is still competitive when k/n is small (i.e., large n).

Quantization Error. We now turn to the quantization error of f-divergences, i.e.,

$$\inf_{|\mathcal{S}|\le k} |D_f(P\|Q) - D_f(P_{\mathcal{S}}\|Q_{\mathcal{S}})|,$$

where the infimum is over all partitions $\mathcal{S}$ of $\mathcal{X}$ of size no larger than k, and $P_{\mathcal{S}}$ and $Q_{\mathcal{S}}$ are the quantized versions of P and Q according to $\mathcal{S}$. We do not assume $\mathcal{X}$ to be discrete, nor do we need Assumption 9 to hold. All the results hold for the Frontier Integral (Property 7) and pointwisely on the divergence frontier (Property 5). Our analysis is inspired by the asymptotic approximation of an f-divergence with increasingly finer partitions (Györfi and Nemetz, 1978, Theorem 6). The key idea behind the proof is shown in Figure 7 and the full proof is given in Appendix B.3.

Proposition 13. *For any two distributions P, Q over $\mathcal{X}$ and any $k \ge 1$, we have*

$$\inf_{|\mathcal{S}|\le 2k} |D_f(P\|Q) - D_f(P_{\mathcal{S}}\|Q_{\mathcal{S}})| \le \frac{f(0) + f^*(0)}{k}\,, \tag{18}$$

where the infimum is over all partitions of $\mathcal{S}$ of size at most $2k$.

Total Error. Combining the bounds on the statistical and quantization errors leads to the following bound for the total estimation error for the Frontier Integral.

Theorem 14. *Assume that $\mathcal{S}_k$ is a partition of $\mathcal{X}$ such that $|\mathcal{S}_k| = k \ge 2$. Then, the total error $\mathbb{E}|\mathrm{FI}(\hat{P}_{\mathcal{S}_k,n}, \hat{Q}_{\mathcal{S}_k,n}) - \mathrm{FI}(P,Q)|$ is upper bounded by*

$$C\big[\,(\alpha_n(P) + \alpha_n(Q))\log n + \beta_n(P) + \beta_n(Q) + |\mathrm{FI}(P,Q) - \mathrm{FI}(P_{\mathcal{S}_k}, Q_{\mathcal{S}_k})|\big]. \tag{19}$$

Moreover, if the quantization error of $\mathcal{S}_k$ satisfies the bound in (18), we have

$$\mathbb{E}|\mathrm{FI}(\hat{P}_{\mathcal{S}_k,n}, \hat{Q}_{\mathcal{S}_k,n}) - \mathrm{FI}(P,Q)| \le C\left[\left(\sqrt{\frac{k}{n}} + \frac{k}{n}\right)\log n + \frac{1}{k}\right]. \tag{20}$$

Algorithm 1 Mauve estimation via vector quantization

```
Input: Samples $\{\boldsymbol{x}_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} P$ and $\{\boldsymbol{x}'_j\}_{j=1}^m \overset{\text{i.i.d.}}{\sim} Q$, quantization size $k$, smoothing constant
    $b$, embedding model $\varphi$, discretization $\Lambda$ of $[0,1]$.
1: $\{\varphi(\boldsymbol{x}_i)\}_{i=1}^n, \{\varphi(\boldsymbol{x}'_j)\}_{j=1}^m \leftarrow \texttt{embed}\left(\varphi, \{\boldsymbol{x}_i\}_{i=1}^n, \{\boldsymbol{x}'_j\}_{j=1}^m\right)$    ▷ Embed the samples
2: $C = \texttt{quantize}\left(\{\varphi(\boldsymbol{x}_i)\}_{i=1}^n, \{\varphi(\boldsymbol{x}'_j)\}_{j=1}^m\right)$    ▷ Cluster embeddings jointly
3: For $l = 1, \dots, k$, set    ▷ Count cluster assignments
       $\hat P^b_{\mathcal{S},n,l} = \frac{1}{n+kb}\left(\sum_{i=1}^n \mathbb{1}\{C(\boldsymbol{x}_i) = l\} + b\right), \hat Q^b_{\mathcal{S},m,l} = \frac{1}{m+kb}\left(\sum_{j=1}^m \mathbb{1}\{C(\boldsymbol{x}'_j) = l\} + b\right)$
4: Compute $\hat{\mathcal{F}}_f(\hat P^b_{\mathcal{S},n}, \hat Q^b_{\mathcal{S},m})$ from (21) for $\lambda \in \Lambda$    ▷ Build the divergence frontier
5: return $\textsc{Mauve}_f(P,Q) \approx \texttt{AUC}\left(\exp\left(-c\,\hat{\mathcal{F}}_f(\hat P^b_{\mathcal{S},n}, \hat Q^b_{\mathcal{S},m})\right)\right)$    ▷ Numerical quadrature
```

Based on the bound in (20), a good choice of k is $\Theta(n^{1/3})$ which balances between the statistical error and the quantization error. This balancing is enabled by the existence of a good vector quantizer with a distribution-free bound in (18). In practice, this suggests a data-dependent vector quantizer using nonparametric density estimators. However, directions such as kernel density estimation (Meinicke and Ritter, 2002; Hegde et al., 2004; Hulle, 1999) and nearest-neighbor methods (Alamgir et al., 2014) have not met empirical success *for vector quantization*, as they suffer from the curse of dimensionality common in nonparametric estimation. In particular, Wang et al. (2005); Silva and Narayanan (2007, 2010) propose quantized divergence estimators but only prove asymptotic consistency and little progress has been made since then. On the other hand, modern data-dependent vector quantization techniques based on deep neural networks can successfully estimate properties of the density from high dimensional data (Sablayrolles et al., 2019; Hämäläinen and Solin, 2020). Theoretical results for those techniques could complement our analysis. We leverage these powerful methods to scale our approach on real data in Section 7. In addition, while nonparametric estimators are not very successful for vector quantization, we can utilize them to estimate the f-divergences directly; we return to this in Section 4.2.

4.1.2 Towards a Practical Algorithm

To develop a practical vector quantization-based estimation procedure for the divergence frontier $\mathcal{F}_f(P,Q)$, we use a data-dependent partitioning $\mathcal{S}$ based on quantizing the samples in some embedding space. The overall procedure is summarized in Algorithm 1.

Recall that we use vector quantization because the support size of real-world text or image distributions is extremely large. We employ embeddings from a pre-trained deep neural network to compute the vector quantization; such deep representations have been shown to capture the important properties of the data across modalities (Zhang et al., 2018; Devlin et al., 2019).

Concretely, we embed the samples using a model $\varphi : \mathcal{X} \to \mathbb{R}^d$ to get $\{\varphi(\boldsymbol{x}_i)\}_{i=1}^n$ and $\{\varphi(\boldsymbol{x}'_j)\}_{j=1}^m$. Then, we jointly quantize the embedded samples to obtain a mapping $C : \mathcal{X} \to$

$[k]$. This induces a partitioning $\mathcal{S} = (S_1, \dots, S_k)$ with $S_l = \{\boldsymbol{x} \in \mathcal{X} \,:\, C(\boldsymbol{x}) = l\}$. For instance, with k-means clustering (Manning and Schütze, 2001; Jurafsky and Martin, 2009), $C(\boldsymbol{x})$ denotes the index l of a cluster center $\boldsymbol{c}_l$ that is closest to embedding $\varphi(\boldsymbol{x})$ in terms of L_2 distance so that each partition $S_l \in \mathcal{S}$ is the Voronoi cell

$$S_l = \left\{ \boldsymbol{x} \in \mathcal{X} \,:\, \|\varphi(\boldsymbol{x}) - \boldsymbol{c}_l\|_2 \le \|\varphi(\boldsymbol{x}) - \boldsymbol{c}_j\|_2 \text{ for } j = 1, \dots, k \right\}.$$

Here, we assume that ties are broken arbitrarily.

The quantized distribution $P_{\mathcal{S}}$ is now computed from the fraction of the points in each partition. For the add-b smoothing, the estimator is

$$\hat{P}^b_{\mathcal{S},n,l} = \frac{1}{n + kb} \left(\sum_{i=1}^{n} \mathbb{1}\{\boldsymbol{x}_i \in S_l\} + b \right), \quad \text{for } l = 1, \dots, k\,.$$

Note that $b = 0$ reduces to the empirical distribution, and this coincides with the approach used in (Pillutla et al., 2021). In this work, we default to Krichevsky-Trofimov smoothing, which corresponds to $b = 1/2$.

Each coordinate of the estimated divergence curve is now an f-divergence of the form $D_f(P_{\mathcal{S},n} \| \lambda \hat{P}^b_{\mathcal{S},n} + (1-\lambda)\hat{Q}^b_{\mathcal{S},m})$ and can be computed by summing over the k coordinates. The full divergence frontier $\mathcal{F}_f(\hat{P}^b_{\mathcal{S},n}, \hat{Q}^b_{\mathcal{S},m})$ is a continuously parameterized curve for $\lambda \in (0,1)$. For computational tractability, we take a discretization Λ of $(0,1)$ and take

$$\hat{\mathcal{F}}_f(P, Q) = \left\{ \big(D_f(Q \| R_\lambda), D_f(P \| R_\lambda) \big) \,:\, \begin{matrix} R_\lambda = \lambda P + (1-\lambda) Q, \\ \lambda \in \Lambda \end{matrix} \right\}. \tag{21}$$

We take a uniform grid $\Lambda = \{1/N, 2/N, \dots, (N-1)/N\}$ with N points. Finally, we approximate $\text{Mauve}_f(P, Q) \approx \text{Mauve}_f(\hat{P}^b_{\mathcal{S},n}, \hat{Q}^b_{\mathcal{S},m})$ using numerical quadrature on the discretized frontier $\hat{\mathcal{F}}_f(\hat{P}^b_{\mathcal{S},n}, \hat{Q}^b_{\mathcal{S},m})$. For FI_f, we can directly estimate $\text{FI}_f(P, Q) \approx \text{FI}_f(\hat{P}^b_{\mathcal{S},n}, \hat{Q}^b_{\mathcal{S},m})$ when a closed-form expression is derived from Property 7 (e.g., for KL and χ^2 divergences).

Computational Complexity. The computational complexity of the overall procedure in Algorithm 1 is dominated by the cost of quantization. The complexity of k-means quantization is $O(Tknd)$, where T is the maximum number of Lloyd's iterations and d is the embedding dimension.

4.2 Estimation via Nearest Neighbors

We now turn to the estimation of the divergence frontier and its summaries by counting the nearest neighbors of each sample. We consider nearest neighbors from the ℓ_2-distance in an embedding space. Given an embedding model $\varphi : \mathcal{X} \to \mathbb{R}^d$, we define a metric ρ on the data space $\mathcal{X}$ as

$$\rho(\boldsymbol{x}, \boldsymbol{x}') = \big\| \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{x}') \big\|_2\,.$$

Let $N_k(\boldsymbol{x})$ denote the set of k-nearest neighbors (under the metric ρ) of $\boldsymbol{x}$ from the set $X \cup X'$ where $X = \{\boldsymbol{x}_i\}_{i=1}^n$ are samples from P and $X' = \{\boldsymbol{x}'_j\}_{j=1}^m$ are samples from Q. Following (Noshad et al., 2017), we estimate the f-divergence $D_f(P \| Q)$ with the estimator

$$\hat{D}_{f,k}(X, X') = 0 \vee \frac{1}{m} \sum_{j=1}^{m} f\left(\frac{|N_k(\boldsymbol{x}'_j) \cap X|/n}{|N_k(\boldsymbol{x}'_j) \cap X'|/m} \right). \tag{22}$$

The intuition behind the estimator is that we expect $|N_k(\boldsymbol{x}'_j) \cap X| \propto P(\boldsymbol{x}'_j)$ and $|N_k(\boldsymbol{x}'_j) \cap X'| \propto Q(\boldsymbol{x}'_j)$, so their ratio (with appropriate normalization)

$$\hat{r}(\boldsymbol{x}'_j) = \frac{|N_k(\boldsymbol{x}'_j) \cap X|/n}{|N_k(\boldsymbol{x}'_j) \cap X'|/m} \tag{23}$$

can be considered an estimate of the likelihood ratio $r(\boldsymbol{x}'_j) := P(\boldsymbol{x}'_j)/Q(\boldsymbol{x}'_j)$. The f-divergence $D_f(P\|Q)$ is them estimated as

$$\hat{D}_{f,k}(X, X') = 0 \vee \frac{1}{m}\sum_{j=1}^{m} f\big(\hat{r}(\boldsymbol{x}'_j)\big)\,. \tag{24}$$

4.2.1 Estimation Error Bounds

Nearest neighbor estimation of f-divergences typically requires continuous distributions on a Euclidean space with densities satisfying certain regularity conditions. To this end, we consider estimation on a noisy version of the problem.

First, we pass from a discrete data space $\mathcal{X}$ to an Euclidean embedding space by taking embeddings from a model $\varphi : \mathcal{X} \to \mathbb{R}^d$. While the pushforward distributions $\varphi_\sharp P$ and $\varphi_\sharp Q$ are now supported on $\mathbb{R}^d$, they are not guaranteed to have a density w.r.t. the Lebesgue measure. To overcome this, we consider smooth these pushforward distributions by convolving them with a Gaussian $\mathcal{N}(0, \sigma^2 I_d)$ to get distributions $P' = \varphi_\sharp P \star \mathcal{N}(0, \sigma^2 I_d)$ and $Q' = \varphi_\sharp Q \star \mathcal{N}(0, \sigma^2 I_d)$. Sampling from the convolved distribution is trivial: $\boldsymbol{u}_i = \varphi(\boldsymbol{x}_i) + \boldsymbol{\xi}_i$ and $\boldsymbol{u}'_j = \varphi(\boldsymbol{x}'_j) + \boldsymbol{\xi}'_j$ are a valid samples from P' and Q' respectively for $\boldsymbol{x}_i \sim P$ and $\boldsymbol{x}'_j \sim Q$ with independent Gaussian noise $\boldsymbol{\xi}_i, \boldsymbol{\xi}'_j \sim \mathcal{N}(0, \sigma^2 I_d)$. We analyze the corresponding version of (22) that is constructed using the ℓ_2 distance between the noisy vectors $\boldsymbol{u}_i, \boldsymbol{u}'_j$. We show that this procedure always underestimates the f-divergence.

Property 15. *For any divergence generator f, we have*

$$D_f(P'\|Q') \le D_f(\varphi_\sharp P\|\varphi_\sharp Q) \le D_f(P\|Q)\,.$$

Further, if the data space $\mathcal{X}$ is discrete and the embedding model is injective, i.e., $\varphi(\boldsymbol{x}) \neq \varphi(\boldsymbol{x}')$ for all distinct $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, then the last inequality hold with equality.

Proof The inequalities are direct applications of the data processing inequality for f-divergences. When φ is injective, we have, $(\varphi_\sharp P)\big(\varphi(\boldsymbol{x})\big) = P(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ and similarly for Q. Therefore, $D_f(\varphi_\sharp P\|\varphi_\sharp Q) = D_f(P\|Q)$ follows from an equality on each term of the summation defining the f-divergence. ∎

The nearest neighbor estimation (22) of $D_f(P'\|Q')$ requires the following assumptions.

Assumption 16. *The smoothed distributions P', Q' have densities p', q' w.r.t. the Lebesgue measure, which satisfy the following:*

(B1) *There exists a $B > 0$ such that we have $1/B \le p'(\boldsymbol{u})/q'(\boldsymbol{u}) \le B$ for all $\boldsymbol{u} \in \mathbb{R}^d$.*

Algorithm 2 MAUVE estimation via nearest-neighbors

```
Input: Samples $X = \{\boldsymbol{x}_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} P$ and $X' = \{\boldsymbol{x}'_j\}_{j=1}^m \overset{\text{i.i.d.}}{\sim} Q$, number of nearest neighbors
    $k$, lower dimension $d'$, embedding model $\varphi$, discretization $\Lambda$ of $[0, 1]$.
1: $\{\varphi(\boldsymbol{x}_i)\}_{i=1}^n, \{\varphi(\boldsymbol{x}'_j)\}_{j=1}^m \leftarrow \texttt{embed}\left(\varphi, \{\boldsymbol{x}_i\}_{i=1}^n, \{\boldsymbol{x}'_j\}_{j=1}^m\right)$    ▷ Embed the samples
2: $U \cup U' = \texttt{PCA}\left(\{\varphi(\boldsymbol{x}_i)\}_{i=1}^n \cup \{\varphi(\boldsymbol{x}'_j)\}_{j=1}^m, d'\right)$    ▷ Joint dimensionality reduction
3: Find $N_k(\boldsymbol{u}) = \texttt{k-NN}(k, \boldsymbol{u}, U \cup U')$ for $\boldsymbol{u} \in U \cup U'$    ▷ Find $k$-nearest neighbors jointly
4: Estimate $\hat{r}(\boldsymbol{u})$ for $\boldsymbol{u} \in U \cup U'$ as    ▷ Estimate the likelihood ratio
       $$\hat{r}(\boldsymbol{u}) = \frac{|N_k(\boldsymbol{u}) \cap U|/n}{|N_k(\boldsymbol{u}) \cap U'|/m}$$
5: Compute $\hat{\mathcal{F}}_{f,k}(P, Q)$ from (25) for $\lambda \in \Lambda$    ▷ Build the divergence frontier
6: return $\text{MAUVE}_{f,k}(P, Q) = \texttt{AUC}\left(\exp\left(-c\,\hat{\mathcal{F}}_{f,k}(P, Q)\right)\right)$    ▷ Numerical quadrature
```

(B2) *The densities p', q' are Hölder continuous with coefficient $\gamma \in (0, 1]$. That is, there exists a constant $H > 0$ such that*

$$|p'(\boldsymbol{u}) - p'(\boldsymbol{u}')| \le H \|\boldsymbol{u} - \boldsymbol{u}'\|_2^\gamma \quad \textit{for all } \boldsymbol{u}, \boldsymbol{u}' \in \mathbb{R}^d \,,$$

and similarly for q'.

The estimator (22) satisfies the following guarantee.

Theorem 17 (Noshad et al. (2017)). *Suppose the smoothed distributions P', Q' satisfy Assumption 16, and the divergence generator f is L-Lipschitz over $[1/B, B]$, where B is from Assumption **(B1)**. Then, the k-nearest neighbor estimator (22) with sample size $m = n$ satisfies*

$$\left|\mathbb{E}[\hat{D}_{f,k}(X, X')] - D_f(P' \| Q')\right| \le O\left(\left(\frac{k}{n}\right)^{\gamma/d} + \frac{1}{k}\right) .$$

The assumption of f being Lipschitz on a restricted domain $[1/B, B]$ follows directly from Assumption **(A2)** with a $\log B$ factor. Thus, this assumption holds for many f-divergences as shown in Table 1. The bound shows that this estimator suffers from the curse of dimensionality, as is common for nonparametric estimators. The two terms of the error are balanced at $k = n^{\gamma/(d+\gamma)}$ and the optimal rate is $n^{-2\gamma/(d+\gamma)}$.

4.2.2 Towards a Practical Algorithm

We note from Theorem 17 that the nearest neighbor estimator (22) suffers from the curse of dimensionality. The embeddings obtained from pre-trained deep nets are extremely high-dimensional, ranging between 10^3 and 10^4 for typical text and image models. We find empirically that a dimensionality reduction step to $d' < 50$ dimensions with principal component analysis (PCA) is crucial for the estimator to work. The overall algorithm is given in Algorithm 2.

As in the case of estimation via quantization, we only consider the points on the divergence frontier at a discretization Λ of $(0, 1)$. We then approximate each coordinate $x(\lambda)$ and $y(\lambda)$ of the divergence frontier for $\lambda \in \Lambda$ by using the nearest neighbor estimator (22). Concretely, this gives us

$$\hat{\mathcal{F}}_{f,k}(P, Q) = \left\{ \left(\hat{D}_{f_\lambda,k}(X, X'), \hat{D}_{f_{1-\lambda},k}(X', X)\right) \,:\, \lambda \in \Lambda \right\}, \tag{25}$$

where f_λ is as defined in Property 5 so that $D_{f_\lambda}(P\|Q) = D_f(P\|\lambda P + (1-\lambda)Q)$. Finally, we estimate $\textsc{Mauve}_f(P, Q)$, $\mathrm{FI}_f(P, Q)$, and $\mathrm{Mid}_f(P, Q)$ from this curve with numerical quadrature or with closed-form expressions when available.

Computational Complexity. The PCA step of Algorithm 2 has time complexity $O(dn^2 + d'd^2)$ while the nearest neighbor search with K-d tree or ball tree structures takes time $O((d' + k)n \log n)$, assuming $n = m$. While both steps can be sped up with approximate randomized algorithms, efficient open-source implementations of exact algorithms are fast enough for problems with a few thousand samples. We use the library Faiss (Johnson et al., 2019) in our experiments in §7.

4.2.3 Extensions and Variants

We could also similarly define a kernel density estimator instead of the nearest neighbor estimator (e.g. Devroye et al., 1996). Given a kernel $\kappa : \mathbb{R}^d \to \mathbb{R}_+$ normalized such that $\int_{\mathbb{R}^d} \kappa(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = 1$, the kernel density estimate of the density of a distribution R using i.i.d. samples $U = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$ is given by

$$g_{\kappa,h,U}(\boldsymbol{u}) = \frac{1}{|U|h^d} \kappa\left(\frac{\boldsymbol{u} - \boldsymbol{u}_i}{h}\right), \tag{26}$$

where h is a bandwidth parameter. A typical choice of kernel is the Gaussian kernel $\kappa(\boldsymbol{z}) = (2\pi)^{-d/2} \exp(-\|\boldsymbol{z}\|_2^2/2)$.

Similar to the nearest neighbor approach, we define the kernel density estimator in the embedding space of a model $\varphi : \mathcal{X} \to \mathbb{R}^d$. We approximate $D_f(P\|Q)$ that of the kernel density estimator using samples $X \sim P^n$ and $X' \sim Q^m$ as $D_f(g_{\varphi(X)}\|g_{\varphi(X')})$, which is in turn estimated using its plug-in estimate

$$\hat{D}_{f,\kappa,h}(X, X') = \frac{1}{m} \sum_{j=1}^{m} f\left(\frac{g_{\kappa,h,\varphi(X)}\big(\varphi(\boldsymbol{x}'_j)\big)}{g_{\kappa,h,\varphi(X' \setminus \{\boldsymbol{x}'_j\})}\big(\varphi(\boldsymbol{x}'_j)\big)}\right). \tag{27}$$

The expectation over Q is approximated by a sample average over X'. The numerator of the term inside $f(\cdot)$ is simply the kernel density estimate (26) of P at $\boldsymbol{x}'_j$ using all n samples from X, while the denominator is the corresponding estimate for Q using the other $m - 1$ samples $X' \setminus \{\boldsymbol{x}'_j\}$. The rest of the estimation procedure is identical to Algorithm 2.

4.3 Estimation via Classification

Here, we consider estimating the likelihood ratio $r(\boldsymbol{x}) := P(\boldsymbol{x})/Q(\boldsymbol{x})$ with a probabilistic classifier such as logistic regression (Sugiyama et al., 2012). The f-divergences can then be estimated from this likelihood ratio.

We first set up a binary classification problem to discriminate between the two distributions P and Q. Concretely, define the class prior as $\mathbb{P}(y=+1) = n/(n+m)$ and $\mathbb{P}(y=-1) = m/(n+m)$ and the class-conditional distribution by $\mathbb{P}(\boldsymbol{x}|y=+1) = P(\boldsymbol{x})$ and $\mathbb{P}(\boldsymbol{x}|y=-1) = Q(\boldsymbol{x})$. By the Bayes rule, the likelihood ratio can equivalently be written as

$$r(\boldsymbol{x}) := \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})} = \frac{\mathbb{P}(y=+1|\boldsymbol{x})}{\mathbb{P}(y=-1|\boldsymbol{x})}\,\frac{\mathbb{P}(y=-1)}{\mathbb{P}(y=+1)}\,.$$

Given a probabilistic classifier that outputs an estimate $\hat\eta(\boldsymbol{x})$ for $\mathbb{P}(y=1|\boldsymbol{x})$, we can estimate the likelihood ratio as

$$\hat r(\boldsymbol{x}) = \frac{m\,\hat\eta(\boldsymbol{x})}{n(1-\hat\eta(\boldsymbol{x}))} = \frac{m}{n}\,\rho(\boldsymbol{x})\,, \tag{28}$$

where $\rho(\boldsymbol{x}) := \hat\eta(\boldsymbol{x})/(1-\hat\eta(\boldsymbol{x}))$ is the odds ratio. We then estimate the f-divergence $D_f(P\|Q)$ using the Monte Carlo estimate

$$\hat D_f(X, X'; \hat p) = \frac{1}{m}\sum_{j=1}^{m} f\left(\hat r(\boldsymbol{x}'_j)\right) = \frac{1}{m}\sum_{j=1}^{m} f\left(\frac{m\,\hat\eta(\boldsymbol{x}'_j)}{n(1-\hat\eta(\boldsymbol{x}'_j))}\right)\,. \tag{29}$$

To train a classifier, we split $X = X_1 \cup X_2$ and $X' = X'_1 \cup X'_2$, train a probabilistic classifier such as a logistic regression model to separate X_1 from X_2 (train set) and evaluate the likelihood ratios on X'_1 and X'_2 (validation set) to estimate the f-divergence.

Practical Considerations. Logistic regression can fail to yield meaningful odds ratio estimates when the two distributions are well-separated. For evaluation of image generative models such as GANs, Lopez-Paz and Oquab (2017) found that neural networks on the pixel space capitalize on artifacts in the generated images, leading to perfect classification and therefore, poor likelihood ratio estimates. To avoid this issue, we employ a linear model on frozen embeddings $\varphi : \mathcal{X} \to \mathbb{R}^d$.

5. Related Work

We focus in this paper on information divergence-based scores to evaluate generative models. While the evaluation process is *post hoc* and external to a generative model, it is worthwhile to mention the increasingly active research area analyzing (classes of) generative models and establishing theoretical results such as statistical consistency, universal approximation, sample complexity; see e.g. (Biau et al., 2021; Schreuder et al., 2021) and references therein. We review the related work on statistical trade-off curves, information divergence-based scores for texts and images, and theoretical results on the statistical estimation of information divergences in mathematical statistics and information theory.

5.1 Divergence Frontiers for Generative Models

Sajjadi et al. (2018) and Kynkäänniemi et al. (2019) both proposed to account for the two types of errors of generative modeling using trade-off curves in the spirit of operation characteristics and precision-recall curves for binary classification and statistical detection (Cortes and Mohri, 2005; Clémençon and Vayatis, 2009; Clémençon and Vayatis, 2010; Flach, 2012).

In an inspiring paper, Djolonga et al. (2020) proposed information divergence frontiers based on Rényi divergences thereby encompassing both (Sajjadi et al., 2018) and (Kynkäänniemi et al., 2019). The authors of (Djolonga et al., 2020) show how to compute the divergence frontiers in special cases such as exponential families. Their exploration of statistical estimation via vector quantization leads to two observations. First, a small quantization size can lead to a bias of optimism, where $D_f(P_{\mathcal{S}} \| Q_{\mathcal{S}}) \le D_f(P\|Q)$ and this gap can be large when $|\mathcal{S}|$ is small. Second, the statistical error from small sample sizes can lead to pessimistic estimates of the divergences. However, (Djolonga et al., 2020) do not provide statistical bounds for vector quantization nor do they analyze statistical properties of divergence frontiers defined using f-divergences. Moreover, the above research does not consider applications to open-ended text generation.

We extend the above line of work, presenting a general framework for estimating divergence frontiers and their statistical summaries for generative models. Theoretically, we provide quantitative upper bounds for both the statistical error and quantization error. Specifically, we show that the statistical error is bounded by $\tilde{O}(\sqrt{k/n})$. Our bounds also demonstrate the interest of using smoothed distribution statistical estimators to account for the missing mass problem. We explore other estimation procedures based on nonparametric nearest-neighbor and kernel density estimation, classifier-based estimation, and parametric Gaussian approximations. We also perform a thorough empirical evaluation and operationalize these scores for large-scale text and image models. Finally, based on our observations, we discuss practical recommendations, to facilitate the application to applied AI domains.

After the publication of the conference paper (Pillutla et al., 2021), subsequent work has corroborated that the original Mauve score compares favorably to other automatic metrics for evaluating neural text (Kour et al., 2022). Pimentel et al. (2023) corroborated the correlation between this score and human judgment. Their empirical analysis shows that a 5-gram estimation of Mauve[3] has a much weaker correlation with human evaluations than the vector quantization procedure used in (Pillutla et al., 2021) (cf. §4.1). Based on this analysis, Pimentel et al. (2023) conclude that the key to the empirical success of Mauve is the vector quantization procedure. The experiments in Section 7 indicate that the reality is much more nuanced. Indeed, we show that the other nonparametric, parametric, and classifier-based estimation of Section 4 can be nearly as effective as vector quantization with the right hyperparameters (§7.3); thus the vector quantization cannot be the driving factor behind Mauve's usefulness as an evaluation metric. We note, however, that vector quantization has several other benefits, including its simplicity and the availability of fast open-source implementations.

Instead, we show that Mauve *requires an embedding of text to vectors to work well in practice*: modern transformer language model embeddings as used in (Pillutla et al., 2021) work well but simple non-contextual GloVe embeddings also work equally well (§7.5). However, estimation from string kernel embeddings[4] (§7.5.4) or direct estimation with language model probabilities (§7.3.3) both fail to quantify previously known trends.

[3] This involves estimating Mauve$(P, Q) \approx$ Mauve$(\hat{P}_{\text{5-gram}}, \hat{Q}_{\text{5-gram}})$ using 5-gram language models $\hat{P}_{\text{5-gram}}, \hat{Q}_{\text{5-gram}}$ fit to samples from P, Q respectively.

[4] An example of a string kernel is the N-gram kernel defined in §7.5.4; this is directly comparable with the N-gram estimation of Mauve in the analysis in (Pimentel et al., 2023).

The original Mauve score has since then been adopted by the language modeling and computational linguistics communities to measure performance and to tune hyper-parameters in diverse language generation settings, including the design of decoding algorithms (Meister et al., 2022; Hewitt et al., 2022; Su et al., 2022; Li et al., 2023; Finlayson et al., 2023), controllable text generation (Yang et al., 2023), architectural innovations (Hu et al., 2022), and differentially private language generation (Mattern et al., 2022; Yue et al., 2023; Kurakin et al., 2023).

Since the publication of the conference paper (Liu et al., 2021), there has been some recent work on the theoretical and algorithmic aspects of divergence frontiers. Verine et al. (2023) give a novel representation of the precision-recall metrics of Sajjadi et al. (2018) as f-divergences. They leverage a classifier-based estimation approach of these f-divergences (similar to §4.3) to optimize image generative models specifically for a given tradeoff between quality and diversity. Kim et al. (2023) propose a variant of generative precision-recall of Kynkäänniemi et al. (2019) that is robust to outliers in the data and their extracted features. They show how it can be computed using a nonparametric kernel density estimator (similar to §4.2.3 but using random projections to evade the curse of dimensionality) in a statistically consistent manner. Cheema and Urner (2023) propose a variant of generative precision-recall and show that a nearest neighbor estimator converges to a well-defined population quantity. As shown in our preliminary conference paper (Liu et al., 2021) and elaborated on in this work, (f-)divergence frontiers can also be estimated in a statistically consistent manner with both vector quantization and nonparametric k-nearest neighbor-based approaches.

5.2 Divergence Measures for Text

Prior measures of similarity/divergence between machine text and human text come in three broad categories: (a) reference-based, (b) statistics-based, and (c) language modeling.

Reference-based metrics evaluate generated text by comparing it with a (small set of) reference text sample(s), rather than comparing distributions over full sequence. These include classical metrics for n-gram matching (Papineni et al., 2002; Lin, 2004; Banerjee and Lavie, 2005), which are designed to capture similarities in the surface form of the generated text and the human references, making them fundamentally ill-suited for open-ended generation. Moreover, it has been shown in (Novikova et al., 2017) that these classical metrics only weakly agree with human judgments.

More recent reference-based metrics are capable of comparisons in a high dimensional embedding space (Shimanaka et al., 2018; Zhang et al., 2020; Sellam et al., 2020; Clark et al., 2019), thereby capturing distributional semantics beyond superficial n-gram statistics. For instance, Moverscore (Zhao et al., 2019) relies on the Word Mover's distance (Kusner et al., 2015), and is an instance of an optimal transport distance (Villani, 2003). Moverscore computes the minimum cost of transforming the generated text to the reference text, taking into account the Euclidean distance between vector representations of n-grams, as well as their document frequencies. The paradigm of reference-based metrics is useful for targeted generation tasks such as translation and summarization, where matching a set of references is paramount. However, this family of metrics is unsuitable for the open-ended generation task where there typically are several plausible continuations for each context and creative generations are desirable. Chan et al. (2022) consider distribution-aware reference-based metrics for conditional generation tasks to account for the diversity in the output space.

Statistics-based metrics compare the model distribution Q with respect to the human distribution P on the basis of some statistic $T(P)$ and $T(Q)$. Property-specific statistics such as the amount of repetition (Holtzman et al., 2020; Welleck et al., 2020b), verifiability (Massarelli et al., 2020), or termination (Welleck et al., 2020a) are orthogonal to Mauve, which provides a summary of the overall gap between P and Q rather than focusing on an individual property. Another statistic is the generation perplexity (Fan et al., 2018; Holtzman et al., 2020), which compares the perplexity of the model text $\boldsymbol{x} \sim Q$ with that of human text $\boldsymbol{x}' \sim P$ under an external model R. We find in Section 7 that generation perplexity fails to correctly capture the effect of the decoding algorithm and the text length. Moreover, it can easily be fooled by an adversarial decoder that generates gibberish text that nevertheless has the right perplexity, as we show in Section 7.2.

Language modeling metrics calculate how (un)likely human text $\boldsymbol{x} \sim P$ is under the model distribution Q, for instance, using the probability $Q(\boldsymbol{x})$. These metrics are related to a single point on the divergence curve, rather than a full summary. Examples include the perplexity of the test set (which is a sample from P) under the model Q and its generalizations to handle sparse distributions (Martins et al., 2020). Unlike the proposed measures, these metrics never see model text samples $\boldsymbol{x}' \sim Q$, so they cannot account for how likely the model text is under the human distribution P. Moreover, they cannot be used for decoding algorithms such as beam search which do not define a token-level distribution.

Automatic metrics have been proposed for specific domains such as generation of dialogues (Tao et al., 2018), stories (Guan and Huang, 2020), and others (Opitz and Frank, 2021). They capture task-specific properties; see the surveys (Celikyilmaz et al., 2020; Sai et al., 2023). In contrast, Mauve compares machine and human text in a domain-agnostic manner. Other related work has proposed metrics that rely on multiple samples for quality-diversity evaluation (Caccia et al., 2020), and Bayesian approaches to compare the distribution of statistics in machine translation (Eikema and Aziz, 2020).

Gehrmann et al. (2023) point out the challenges involved in designing good automatic evaluation metrics with a focus on directed generation tasks. They outline many suggestions including continuously updated suites of datasets, documentation, and benchmarks, as well as a multi-dimensional evaluation with each metric focusing on a small yet more precisely defined scope. Liang et al. (2023) advocate for a multi-metric approach for evaluating generated language, going beyond quality and considering specific attributes such as toxicity and bias.

Non-Automatic Metrics. HUSE (Hashimoto et al., 2019) aims to combine human judgments of Type I errors with Type II errors measured using perplexity under Q. Due to the costs of human evaluation, we consider HUSE and other metrics requiring human evaluation, such as single-pair evaluation, complementary to the proposed automatic measures. As a separate technical caveat, it is unclear how to use HUSE for sparse Q that assigns zero probability to a subset of text, which is the case with state-of-the-art decoding algorithms (Holtzman et al., 2020; Martins et al., 2020; Meister et al., 2022).

5.3 Divergence Measures for Images

Evaluation of generative models is an active area of research in computer vision, where implicit models including generative adversarial networks (Goodfellow et al., 2014) preclude even basic divergence evaluations based on test-set log-likelihoods. The popular Inception Score (Salimans et al., 2016) is based on large-scale supervised classification tasks; it is unclear how to adapt this score to other modeling domains, such as open-ended text generation. The Fréchet Inception Distance (Heusel et al., 2017; Semeniuta et al., 2018) and its unbiased counterpart, the Kernel Inception Distance (Bińkowski et al., 2018) are both used for evaluating generative models, but, unlike divergence frontier methods, do not take into account trade-offs between different kinds of errors between the learned and the reference distribution. We find in Section 7.2 that the Fréchet distance adopted to the text setting fails to capture the dependence on the text length, while our proposed approach can. We note that this sequential temporal aspect is absent in the image modality. An exploration of this property of Fréchet distance and Mauve in other sequential modalities such as video (Unterthiner et al., 2018) and speech (Kilgour et al., 2019) is an interesting direction for future work.

5.4 Statistical Estimation of Information Divergences

A closely related problem is the estimation of functionals of discrete distributions; see (Verdú, 2019) for an overview. In particular, the estimation of KL divergences has been studied in both fixed and large alphabet regimes (Cai et al., 2006; Zhang and Grabchak, 2014; Bu et al., 2018; Han et al., 2020). An important result from this line of research is that the minimax quadratic risk of the naïve plug-in estimator is infinite (Bu et al., 2018). The main challenge arises from the missing mass phenomenon (Good, 1953) which is especially prominent in the large alphabet regime. This challenge can be addressed by applying add-constant smoothing (Krichevsky and Trofimov, 1981; Braess and Sauer, 2004) to the empirical distribution estimator and requiring the two distributions to have a bounded density ratio. Our results also utilize add-constant smoothing without the need for the boundedness assumption. Other choices of estimators include the Good-Turing (Good, 1953) and the absolute discounting (Falahatgar et al., 2017) estimators.

On the practical side, there is a new line of successful work that uses deep neural networks to find data-dependent vector quantization to estimate information-theoretic quantities from samples (Sablayrolles et al., 2019; Hämäläinen and Solin, 2020). Our experimental results also rely on such data-dependent vector quantizers.

There exists a rich literature on statistical estimation of f-divergences using other methods. Nonparametric estimation of f-divergences via nearest-neighbor and kernel density estimation was studied in (Póczos et al., 2011; Moon and Hero III, 2014; Kandasamy et al., 2015; Noshad et al., 2017), to name a few. The variational expression for f-divergences was leveraged for optimization-based estimation in (Nguyen et al., 2010; Sreekumar and Goldfeld, 2022). Estimation under structural assumptions satisfied in applications such as autoencoders was considered in (Rubenstein et al., 2019). While not directly related to statistical estimation, a general optimization-based methodology to derive sharp inequalities between various f-divergences was given in (Guntuboyina et al., 2014). In contrast, we focus on vector quantization-based estimation while empirically comparing them to ap-

Task Domain	Model	Finetuning	Dataset	Prompt Length	Max. Generation Length	Number of Generations
Web text	GPT-2 (all sizes)	Pretrained	Webtext	35 tokens	1024 tokens	5000
News	Grover (all sizes)	Pretrained	RealNews	varying	1024 tokens	5000
Stories	GPT-2 medium	Finetuned	WritingPrompts	50 tokens	512 tokens	5000

Table 3: Dataset and task summary for open-ended text generation. Note that 1024 tokens correspond to ~ 750 words on average.

proaches based on nonparametric estimators, classifier-based estimation, and parametric approximation.

6. Experiments: Setup

We consider open-ended text generation tasks, where the model has to generate text in continuation of a given text prompt. The open-endedness of the task is reflected in the relative lengths of the prompt and the generation: the prompt is often quite short (35 to 50 tokens), while the generation is 10× to 30× longer (approximately 500 to 1000 tokens).

6.1 Task Domains and Models

We consider three different text domains: web text, news, and stories. For each domain, we consider generation with size-based variants of transformer language models. See Table 3 for a summary.

Web Text Generation. The goal of this task is to generate articles from the publicly available analogue of the Webtext dataset[5] using pre-trained GPT-2 models for various sizes (Radford et al., 2019; Brown et al., 2020). At generation time, we use as prompts the first 35 tokens of each of the 5000 articles from the Webtext test set, keeping the maximum generation length to 1024 tokens (which corresponds, on average, to around 750 words). For comparison with human text, we use the corresponding human-written continuations from the test set (up to a maximum length of 1024 tokens).

News Generation. Under this task, the goal is to generate the body of a news article, given the title and metadata (publication domain, date, author names). We use a left-to-right transformer language model, Grover (Zellers et al., 2019), which is similar to GPT-2 but tailored to generating news by conditioning on the metadata of the article as well. Our generations rely on pre-trained Grover architectures of various sizes. The generation prompt comprises the headline and metadata of 5000 randomly chosen articles from the "April2019" set of the RealNews dataset (Zellers et al., 2019), and the maximum article length was 1024 tokens. We reuse the publicly available Grover generations[6] for our evaluation.

Story Continuation. Given a situation and a (human-written) starting of the story as a prompt, the goal of this task is to continue the story. Here, we use a GPT-2 medium model fine-tuned for one epoch on the WritingPrompts dataset (Fan et al., 2018). We use

[5] `https://github.com/openai/gpt-2-output-dataset`

[6] available at `https://github.com/rowanz/grover/tree/master/generation_examples`

as generation prompts the first 50 tokens of 5000 randomly chosen samples of the test set of WritingPrompts. The model generations are allowed to be up to 512 tokens long. The corresponding test examples, truncated at 512 tokens are used as samples from P.

6.2 Decoding Algorithms

We consider three common decoding algorithms described in Section 2.1.

(a) **Greedy decoding** selects the most likely next token $x_{t+1} = \arg\max_{x \in \mathcal{V}} \hat{P}(x \mid \boldsymbol{x}_{1:t})$ and is representative of a broader class of approximate likelihood maximization decoding algorithms.

(b) **Ancestral sampling** samples directly from the language model's per-step distributions as $x_{t+1} \sim \hat{P}(\,\cdot \mid \boldsymbol{x}_{1:t})$, and generates unbiased samples from the model distribution.

(c) **Nucleus sampling** (Holtzman et al., 2020) samples from top-p truncated per-step distributions, $x_{t+1} \sim \hat{Q}_{\mathrm{nuc},p}(\,\cdot \mid \boldsymbol{x}_{1:t})$ as defined in Equation (2).

Greedy decoding attempts to find text that approximately maximizes its likelihood under the model. While such algorithms are highly successful for directed text generation tasks such as translation, they produce highly degenerate repetitive text in the open-ended setting. While ancestral sampling produces unbiased samples from the model distribution, it also has been found to generate degenerate text (Holtzman et al., 2020), ostensibly because the model is imperfect, especially in the low-probability tail of the next-token distribution. Nucleus sampling attempts to fix this by truncating the tail and is representative of the broader class of truncated sampling methods that are now widely considered state-of-the-art. We vary the nucleus parameter $p \in \{0.9, 0.92, 0.95, 0.99\}$ for web text generation and story continuation, and $p \in \{0.9, 0.92, 0.94, 0.96, 0.98\}$ for news generation.

In addition, we also consider the following decoding algorithms:

(d) **Beam search** is a more sophisticated approximate likelihood maximization algorithm that maintains a set of b promising prefixes. At each time step, all possible one-token continuations of the current b prefixes are considered and the top b of them are retained.

(e) **Locally typical sampling** (Meister et al., 2022) is a truncation sampling method, which we use as a representative of recent truncation-based decoding algorithms such as Mirostat (Basu et al., 2021) and η-sampling (Hewitt et al., 2022). Locally typical sampling with hyperparameter $\tau \in (0,1)$ samples the next token from the truncated vocabulary

$$V_{\mathrm{typ},\tau} = \arg\min_{V'} \left\{ \sum_{x \in V'} \left| \log \hat{P}(x|\boldsymbol{x}_{1:t}) + H\big(\hat{P}(\cdot|\boldsymbol{x}_{1:t})\big) \right| \,:\, \sum_{x \in V'} \log \hat{P}(x|\boldsymbol{x}_{1:t}) \ge \tau \right\}$$

of the language model $\hat{P}$, where $H(p) = -\sum_{x \in V} p(x) \log p(x)$ is the Shannon entropy. This is a set that covers at least τ-fraction of the probability mass but also has log probabilities that are as close to the conditional entropy as possible. The samples are obtained by sampling from this truncated distribution as

$$Q_{\mathrm{typ},\tau}(x_{t+1} \mid \boldsymbol{x}_{1:t}) = \begin{cases} \frac{1}{Z}\,\hat{P}(x_{t+1} \mid \boldsymbol{x}_{1:t}), & \text{if } x_{t+1} \in V_{\mathrm{typ},\tau}, \\ 0, & \text{else}, \end{cases}$$

where Z is a normalizing constant.

(f) **Adversarial perplexity sampling** is designed to generate low-quality text that nevertheless matches the perplexity of human text. Adversarial perplexity sampling proceeds in two phases: (1) we generate the first 15% of tokens in a sequence uniformly at random from the vocabulary, and (2) we generate the remaining tokens greedily to make the running perplexity of the generated sequence as close as possible to the perplexity of human text.

6.3 Baseline Metrics

We compare the proposed measures to the following automatic evaluation metrics used previously to evaluate open-ended generation.

- **Generation Perplexity (Gen. PPL.)**: We compute the perplexity of the generated text under the GPT-2 large model. A common heuristic is to match
- **Zipf Coefficient**: we report the slope of the best-fit line on the log-log plot of the rank versus unigram frequency plot. Note that the Zipf coefficient only depends on unigram count statistics and is invariant to, for instance, permuting the generations. We use the publicly available implementation of (Holtzman et al., 2020).[7]
- **Repetition Frequency (Rep.)**: The fraction of generations which devolved into repetitions. Any generation that contains at least two contiguous copies of the same phrase of any length appearing at the end of a phrase is considered a repetition. We consider repetitions at the token level. This metric is useful to quantify degenerate repetitiveness that sometimes comes up with neural text (e.g., with greedy decoding).
- **Distinct-n**: The fraction of distinct n-grams from all possible n-grams across all generations. We use $n = 4$. This is a measure of how diverse the generated text is.
- **Self-BLEU**: Self-BLEU is calculated by computing the BLEU score of each generation against all other generations as references. We report the Self-BLEU using 4-grams. This operation is extremely expensive, so we follow the protocol of (Holtzman et al., 2020): sample 1000 generations and compute the BLEU against all other 4999 generations. A lower Self-BLEU score implies higher diversity.
- **Discriminator Accuracy**: We train a binary classifier to classify text as human or not. A smaller discrimination accuracy means that model text is harder to distinguish from human text. A separate classifier is trained for each model and decoding algorithm pair. For the story continuation task, we train a classification head on a frozen GPT-2 large model using the logistic loss. We use 25% of the data as a test set and the rest for training; a regularization parameter is selected with 5-fold cross-validation. For the news dataset, we follow the protocol of (Zellers et al., 2019), i.e., a Grover large model finetuned with a binary classification head.

Apart from discriminator accuracy, every other metric quantifies a property $T(Q)$ of the distribution Q of the generated text. This number makes sense only in comparison to the corresponding quantity $T(P)$ of the human text distribution P. For each of these, we use $|T(Q) - T(P)|$ as a measure of the gap between P and Q.

[7] https://github.com/ari-holtzman/degen/blob/master/metrics/zipf.py

6.4 Human Judgements and Evaluation of Automatic Metrics

An effective metric should yield judgments that correlate highly with human judgments, assuming that human evaluators represent a gold standard.[8] We evaluate how the quality judgments of the proposed measures correlate with human quality judgments. In our study, a quality judgment means choosing a particular (model, decoder) setting based on the resultant generations.

Evaluation Protocol. Since our goal is to measure the gap between a model text distribution Q and a human text distribution P, we employ a pairwise setup for human evaluations. At each round, an annotator receives a context and continuations from two different (model, decoder) settings, and selects the continuation they found more (a) human-like, (b) interesting, and (c) sensible on a 5-point Likert scale. Our interface for collecting annotations is shown in Figure 24 of Appendix E.

We collect these annotations for web text generation with 8 different (model, decoder) settings plus a ninth setting for human-written continuations. Each setting is a GPT-2 model size paired with either ancestral or nucleus sampling. This gives us a total of 36 pairs of settings. Given the known difficulties with human evaluation of longer texts (Ippolito et al., 2020), we use a maximum completion length of 256 tokens. We obtain 90 preference ratings for each pair of settings, coming from a total of 214 crowd-workers from the Amazon Mechanical Turk platform. The evaluators were paid USD 0.40 per evaluation based on an estimated wage of USD 16 per hour.

Pairwise Scores to a Ranking. We convert these pairwise preferences to a ranking by fitting a Bradley-Terry model (Marden, 1995), a parametric model used to predict the outcome of a head-to-head comparison. In particular, we obtain a score w_i for each setting i so that the log odds of humans preferring setting i to setting j in a head-to-head comparison is given by the difference $w_i - w_j$.

Evaluation of Automatic Metrics. Consider an automatic metric M with mean values $\boldsymbol{a} = (a_1, \dots, a_n)$ and standard deviations $\boldsymbol{s} = (s_1, \dots, s_n)$ across n different (model, decoder) pairs, where the mean and standard deviation is over repetitions with multiple random seeds. We assume that higher values of the metric mean that the text is closer to human text. Let $\boldsymbol{h} = (h_1, \dots, h_n)$ denote the Bradley-Terry coefficients obtained from the human evaluation protocol designed above. We evaluate the automatic metric M by comparing the ranking it induces over the (model, decoder) pairs to that obtained by the human evaluation using the Spearman rank correlation.

In order to account for the standard deviation of the metric, we define the **worst-case Spearman rank correlation** between $a_1 \pm s_1, \dots, a_n \pm s_n$ with $\boldsymbol{h} = (h_1, \dots h_n)$ as

$$\rho_{\min}(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{h}) = \min_{\sigma_1, \dots, \sigma_n \in \{-1,1\}^n} \rho\big((a_i + \sigma_i s_i)_{i=1}^n, \boldsymbol{h}\big) \,, \tag{30}$$

where $\rho(\boldsymbol{a}, \boldsymbol{h})$ denotes the Spearman rank correlation between $\boldsymbol{a}$ and $\boldsymbol{h}$. The end result is a correlation score in $[-1, 1]$, with higher values meaning that quality judgments using the

[8] While recent work has shown that human evaluation might not always be consistent (Clark et al., 2021; Karpinska et al., 2021; Gehrmann et al., 2023), human judgments continue to be the gold standard for evaluating open-ended text generation.

Metric	Task	Gen. PPL	Zipf Coef.	REP	Distinct-4	Self-BLEU	Mauve*
Human-like/BT	Web text	0.810	0.762	−0.500	0.738	0.500	**0.857**
Interesting/BT	Web text	0.643	0.405	−0.571	0.524	0.262	**0.714**
Sensible/BT	Web text	0.738	0.643	−0.476	0.595	0.452	**0.762**
Disc. Acc.	News	0.468	0.595	0.792	0.653	0.516	**0.956**
Disc. Acc.	Stories	0.690	0.762	0.190	0.833	**0.905**	**0.905**

Table 4: Correlation of various automatic metrics with human judgments when available, and the accuracy of a trained discriminator otherwise. "BT" denotes the Bradley-Terry score for a pairwise human evaluation. We show the worst-case Spearman rank correlation defined in (30) for the BT scores. Boldfaced/highlighted numbers indicate the highest correlation in each row.

automatic metric correlate with quality judgments made by human evaluators up to one standard deviation from the randomness of sampling.

6.5 Hyperparameters

By default, we summarize the divergence frontier with Mauve$_{\mathrm{KL}}$ computed using k-means vector quantization (Algorithm 1) with $k = 500$ buckets. Following the discussion in Section 4.1, we use the Krichevsky–Trofimov (add-1/2) smoothing. This is different from the default setting of (Pillutla et al., 2021), where the empirical estimator is used instead (with the other hyperparameters remaining the same). To make this distinction clear, we refer to the version computed by the smoothed estimator as Mauve$^\star_{\mathrm{KL}}$ and the original version of (Pillutla et al., 2021) as Mauve$_{\mathrm{KL}}$ (or Mauve* and Mauve respectively when the KL divergence is clear from the context). We compare this choice with different estimation methods in Section 7.3 and different divergence frontier summaries in Section 7.4.

7. Experimental Results

We present the main experimental results in this section. We start by comparing the rankings induced by Mauve to that of the human evaluators in Section 7.1. Next, we demonstrate in Section 7.2 that the proposed measures can quantify how the properties of the generated text vary with model size, decoding algorithms, and text length. Then, we compare in Section 7.3 the different statistical estimation methods discussed in Section 4. We perform a detailed comparison of various f-divergence and optimal transport-based alternatives in Section 7.4. We demonstrate the effect of the embedding model in Section 7.5, and explore the applicability of generative precision-recall (Kynkäänniemi et al., 2019), originally proposed for the vision modality, to the natural language modality in Section 7.6. Finally, we go beyond the language domain to show how the proposed methods can be useful in the vision modality in Section 7.7.

7.1 Comparison to Human Evaluation

We now compare the ranking induced by the proposed measure to that of the human evaluation scores.

Correlation with Human Judgments. Table 4 shows the correlation between human judgments and five automatic evaluation metrics obtained using our evaluation protocol on the web text domain. Mauve correlates highly with human judgments of how human-like (0.857), interesting (0.714), and sensible (0.762) the machine text is. Mauve's correlations with human judgments are substantially higher than those for the other automated measures; for instance, the commonly used generation perplexity has correlations that are 0.810, 0.643, and 0.738 respectively. The results suggest that the proposed measures may act as an effective, automatic surrogate for costly human judgments.

Correlation with Learned Discriminators. We also measure the quality of generations by how well a trained model (a discriminator) can distinguish between real and generated text (Lopez-Paz and Oquab, 2017). We report the test accuracy of a binary classifier trained to discriminate between machine and human text; a lower discrimination accuracy implies that the generation is harder to distinguish from human text. We report the accuracy of Grover-large as the discriminator for the news generations as it produced the highest discrimination accuracy (Zellers et al., 2019) while we use GPT-2 large for the story domain. As seen in Table 4, Mauve correlates the highest with the discrimination accuracy (0.956 for news and 0.905 for stories) among all comparison measures. Computing the discrimination accuracy for each (model, decoder) pair requires fine-tuning a separate model, which is particularly expensive for large models. The proposed measures, on the other hand, do not require any training when computed using vector quantization.

Disagreements between Mauve and Human Judgements. Table 5 gives the values of Mauve and the Bradley-Terry coefficients of the human evaluation for how human-like the text is. Human evaluators find GPT-2 xl with ancestral sampling (BT score of 8.97) to produce text that is more human-like than GPT-2 medium with nucleus sampling (BT score of -3.43), while their Mauve scores are 0.908 and 0.936 respectively. Similarly, Mauve finds GPT-2 large with ancestral sampling to be worse than GPT-2 small with nucleus sampling, while human evaluators disagree. Mauve agrees with the human evaluators on all other pairwise comparisons.

7.2 Quantifying the Effect of Model Size, Decoding, Text Length

To study the effectiveness of the proposed measures for comparing text distributions, we first examine how they quantify known properties of generated text: a good metric should meet expected behavior that is known from existing research on each property. Specifically, we investigate how Mauve behaves under changes in model size, decoding algorithm, and generation length. We give the results of web text generation in Table 5; the corresponding results for the other domains can be found in Appendix D.

Effect of the Model Size. Scaling the model size has been a critical driver of recent advances in natural language processing, with larger models leading to better language modeling and higher-quality open-ended generation. An effective metric should capture the relationship between model size and generation quality, which we verify with human evaluations.

We see from Table 5 that Mauve increases as the model size increases, agreeing with the human evaluation and the expectation that larger models should have higher quality

GPT-2 Size	Decoding	Gen. PPL	Zipf Coef.	Rep.	Distinct-4	Self-BLEU	Human(↑)	Mauve* (↑)
small	Sampling	$101.880_{0.627}$	$0.926_{0.001}$	$0.001_{0.000}$	$0.941_{0.001}$	$0.327_{0.003}$	−27.52	$0.655_{0.018}$
	Greedy	1.224	1.037	0.942	0.072	$0.465_{0.000}$	–	0.019
	Nucleus, 0.9	$23.788_{0.144}$	$1.012_{0.002}$	$0.010_{0.001}$	$0.859_{0.002}$	$0.436_{0.004}$	−15.78	$0.906_{0.005}$
	Adversarial	**12.554**	1.073	0.006	0.365	0.525	–	0.043
medium	Sampling	$129.263_{0.798}$	$0.872_{0.001}$	$0.001_{0.000}$	$0.953_{0.001}$	$0.281_{0.002}$	−30.77	$0.446_{0.010}$
	Greedy	1.241	0.978	0.903	0.091	0.415	–	0.024
	Nucleus, 0.9	$21.073_{0.134}$	$\mathbf{0.957}_{0.001}$	$0.005_{0.001}$	$\mathbf{0.884}_{0.001}$	$\mathbf{0.402}_{0.003}$	−3.43	$0.936_{0.004}$
	Adversarial	**12.554**	1.006	0.005	0.381	0.444	–	0.044
large	Sampling	$30.080_{0.196}$	$0.930_{0.002}$	$\mathbf{0.002}_{0.001}$	$0.916_{0.001}$	$0.358_{0.001}$	−6.93	$0.878_{0.008}$
	Greedy	1.232	0.983	0.881	0.100	0.413	–	0.026
	Nucleus, 0.95	$13.499_{0.058}$	$0.967_{0.002}$	$0.006_{0.001}$	$0.870_{0.001}$	$0.412_{0.002}$	12.55	$0.952_{0.002}$
	Adversarial	**12.554**	0.965	0.005	0.395	0.429	–	0.035
xl	Sampling	$31.886_{0.447}$	$0.930_{0.001}$	$0.002_{0.001}$	$0.913_{0.001}$	$0.360_{0.003}$	8.97	$0.908_{0.005}$
	Greedy	1.278	0.975	0.859	0.115	0.417	–	0.033
	Nucleus, 0.95	$14.143_{0.043}$	$0.966_{0.002}$	$0.005_{0.000}$	$0.868_{0.001}$	$0.413_{0.002}$	**15.66**	$\mathbf{0.955_{0.004}}$
	Adversarial	**12.554**	0.986	0.005	0.397	0.448	–	0.057
Human	n/a	12.602	0.952	0.002	0.878	0.382	47.25	–

Table 5: Automatic metrics across different model sizes and decoding approaches for web text generations. Subscripts indicate the standard deviation across 5 runs for the sampling-based methods; greedy decoding, being deterministic, always returns the same value for a given model. For nucleus sampling, we show the best hyperparameter value from $\{0.9, 0.92, 0.95, 0.99\}$ as per Mauve. The column "Human" gives the Bradley-Terry score obtained from how human-like the text is (Section 6.4). Boldfaced numbers indicate the best performance according to the metric, or closest to the human reference, when applicable.

Decoding	Greedy	Beam		Beam + no 4-gram repeat		Ancestral	Nucleus
		$b = 4$	$b = 8$	$b = 4$	$b = 8$		
Mauve*	0.019	0.040	0.049	0.438	0.415	$0.655_{0.021}$	$\mathbf{0.906_{0.005}}$

Table 6: Beam search with beam sizes $b = 4, 8$ (with and without allowing 4-gram repetitions) versus other decoding algorithms of Table 5 for web text generation with GPT-2 small. The subscript denotes the standard deviation over 5 random seeds and is omitted for the deterministic greedy decoding and beam search.

Decoding	Locally Typical Sampling						Nucleus
	$\tau = 0.2$	$\tau = 0.5$	$\tau = 0.7$	$\tau = 0.9$	$\tau = 0.95$	$\tau = 0.99$	
Mauve*	$0.862_{0.012}$	$0.896_{0.005}$	$0.88_{0.01}$	$0.939_{0.009}$	$\mathbf{0.950}_{0.005}$	$0.914_{0.007}$	$\mathbf{0.952}_{0.003}$

Table 7: Comparing locally typical sampling (Meister et al., 2022) to nucleus sampling ($p = 0.95$) with Mauve for web text generations from GPT-2 large. The subscript denotes the standard deviation over 5 random seeds.

generations. The widely-used generation perplexity, however, incorrectly rates the large model's text as better than the xl model. In this case, human evaluators rate generations from the small model better than those from the medium model. Interestingly, Mauve and Gen. PPL. both identify this relationship, agreeing with the human ratings, in contrast to the other automatic metrics we surveyed.

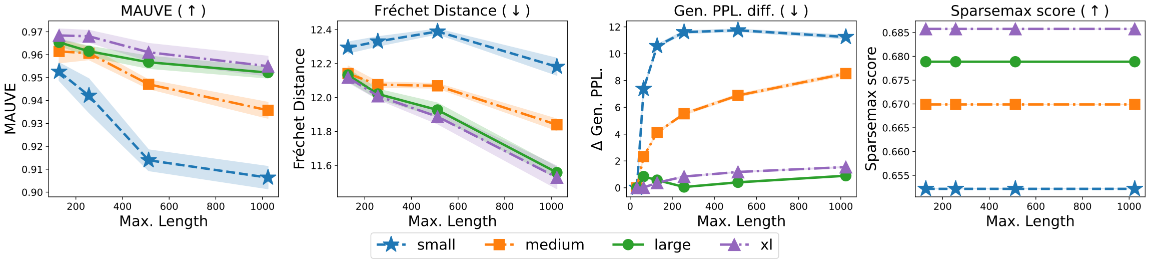


Figure 8: Generation quality versus maximum generation length according to Mauve and three alternative measures (web text, GPT-2). Mauve is the only comparison measure that identifies that generation quality decreases monotonically with increasing text length. The shaded area shows one standard deviation over generations from 5 random seeds.

Effect of the Decoding Algorithm. Recent work has identified two clear trends in open-ended text generation with standard autoregressive models: (1) using greedy decoding results in repetitive, degenerate text (Holtzman et al., 2020; Welleck et al., 2020b,a); (2) nucleus sampling (and related truncated sampling methods) with the right hyperparameter yields higher quality text than ancestral sampling (Fan et al., 2018; Holtzman et al., 2020). An effective measure should thus indicate the quality relationship greedy $\prec$ ancestral $\prec$ nucleus.

We see from Table 5 that Mauve correctly identifies the expected quality relationship, assigning the lowest quality to greedy decoding for the xl model followed by ancestral sampling, and the highest quality to nucleus sampling for all model sizes — these values are $0.016, 0.882, 0.940$ respectively for the xl model. Other commonly used metrics fail to identify this relationship: generation perplexity rates the highly degenerate greedy-decoded text as better than ancestral sampling (a difference of 11.324 w.r.t. the human perplexity vs. 19.284). Furthermore, generation perplexity falls victim to the adversarial decoder that produces gibberish text. Mauve, on the other hand, rightly rates it poorly.

We see in Table 6 that Mauve identifies degeneracy of beam search, thus quantifying the qualitative observations of Holtzman et al. (2020). Next, Table 7 shows that locally typical sampling produces text that is comparable in its Mauve score to nucleus sampling and outperforms other decoding algorithms, echoing the results of Meister et al. (2022).

Effect of the Generation Length. Although large transformer-based models can generate remarkably fluent text, it has been observed that the quality of generation deteriorates with text length: as the generation gets longer, the model starts to wander, switching to unrelated topics and becoming incoherent (Rashkin et al., 2020). As a result, an effective measure should indicate lower quality (e.g. lower Mauve) as generation length increases.

Figure 8 shows Mauve as the generation length increases, along with three alternative metrics: generation perplexity, sparsemax score (Martins et al., 2020), and Fréchet distance (Heusel et al., 2017; Semeniuta et al., 2018). Mauve reflects the desired behavior, showing a decrease in quality as generation length grows, with the trend consistent across model sizes. The other three metrics, however, show less favorable trends. Fréchet distance indicates *improving* quality as the length increases, while generation perplexity shows non-monotonic quality trends for the small and large models. Finally, language modeling

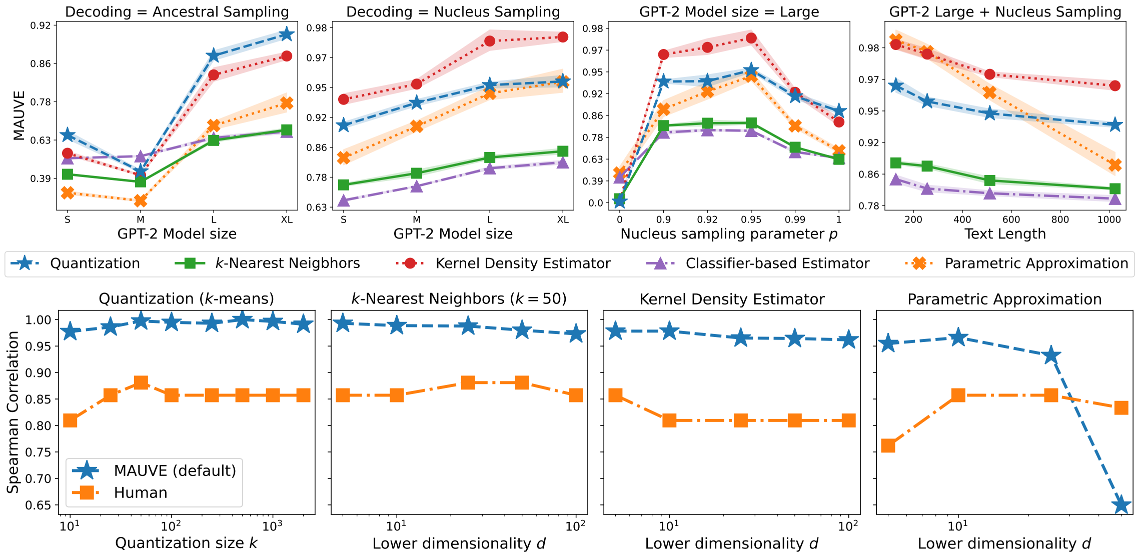


Figure 9: Computing Mauve using different estimation procedures of Section 4. **Top row**: Trends from varying model size, decoding algorithm, and text length. **Bottom row**: Effect of estimation hyperparameters on the correlation with the default setting of Mauve (vector quantization with $k = 500$) and human evaluations from Table 5. These correlations for the classifier-based estimator are **0.979** and **0.857** respectively.

metrics such as the sparsemax score (Martins et al., 2020) remain constant, since they do not depend on the samples generated.

7.3 Comparison of Statistical Estimation Methods

We now compare different methods of estimating Mauve from Section 4 as well as direct estimation from model probabilities.

7.3.1 Comparison of Vector Quantization with Other Approximations

We now compare the different statistical estimation methods from Section 4: vector quantization (Algorithm 1 with Krichevsky–Trofimov smoothing, our default), nearest neighbor estimation (Algorithm 2), kernel density estimation (Algorithm 2 modified as in Section 4.2.3), classifier-based estimation (Section 4.3), and parametric approximation (Appendix C). We also compare these to the direct estimation of Mauve based on model probabilities. We perform these comparisons for the web text domain.

Hyperparameters. The non-parametric nearest neighbor and kernel density estimators and the parametric approximation require the $d = 1024$ dimensional embeddings to be projected into a small m-dimensional subspace. We use the first m principal components of the embeddings. Empirically, we find that the monotonicity property $\mathrm{KL}(P \| \lambda_1 P + (1-\lambda_1)Q) \le \mathrm{KL}(P \| \lambda_2 P + (1-\lambda_2)Q)$ for $\lambda_1 \ge \lambda_2$ can fail to hold in the non-parametric and parametric estimates if $m > 100$. This is a manifestation of the well-known curse of dimensionality for non-parametric estimation and the Monte-Carlo estimation (Equation (46) in Appendix C) required by our parametric approximation. Practically, the failure of the monotonicity property makes it challenging to estimate the area under the curve. We employ a ℓ_2-regularization term to the classifier-based estimator and found the results to be robust to the choice of the regularization parameter in the range $1/n$ to $10^{-3}/n$ where n is the number of samples. We use a different scaling constant c within the exponential (cf. (5)) for each method: $c = 5$ for vector quantization, $c = 10$ for nearest neighbor and kernel density estimation, $c = 2.5$ for classification, and $c = 1$ for the Gaussian approximation. Note that this does not change the induced rankings.

Results. The results are given in Figure 9. We see that each of the estimation methods can identify most of the trends of Section 7.2. As a notable exception, the classifier-based estimate fails to identify the trend that the GPT-2 small model with ancestral sampling is better than the medium one (cf. Table 5). Notably, the parametric approximation identifies the correct dependence on the text length while the parametric approximation of the optimal transport cost, namely the Fréchet distance fails to capture this trend (cf. Figure 8). Interestingly, $m = 5$ or 10 principal components of the embeddings allow us to capture the trends with respect to the model size, decoding algorithms, and text length.

Correlation Analysis. We note that each estimation method exhibits a high Spearman rank correlation with the default vector quantization approach of 0.95 to 1.0 and a worst-case Spearman correlation of at least 0.857 with the human evaluations *for the best hyperparameter values*. We find that the parametric approximation is not robust to the number m of principal components — its performance steeply drops off at $m = 100$.

Pros and Cons of the Estimation Methods. All the tested estimation methods are consistent with each other, demonstrating the versatility of Mauve's recipe of estimating information divergences from vector embeddings of data. However, there are some minor differences. First, the k-nearest neighbor and classifier-based estimators report a tie between nucleus sampling with $p = 0.9$ and $p = 0.95$. In contrast, the vector quantization approach ranks $p = 0.95$ as better than $p = 0.9$; this is also the case with the Gen. PPL. baseline. Second, the non-parametric nearest neighbor and kernel density estimators, as well as the parametric Gaussian approximation require extreme dimensionality reduction, which makes it important to select the lower dimension correctly. In contrast, the quantization performance is more robust to its hyperparameter (the quantization size k). Thus, we recommend the vector quantization approach as a reliable default as it is relatively computationally inexpensive and does not require much hyperparameter tuning.

7.3.2 Effect of Smoothing on Vector Quantization-Based Estimation

We now analyze the effect of smoothing on vector quantization-based estimation. Table 8 compares vector quantization (Algorithm 1) with and without the Krichevsky–Trofimov smoothing. Their Spearman rank correlations are 1.0, meaning that they induce the same ranking. We note that their numerical values can be different, depending on the number of empty bins.

Recall the computation pipeline of Algorithm 1: we jointly quantize the embedded samples $\{\varphi(\boldsymbol{x}_1, \ldots, \varphi(\boldsymbol{x}_n)\}$ and $\{\varphi(\boldsymbol{x}'_1), \ldots, \varphi(\boldsymbol{x}'_m)\}$ from P and Q respectively with k-means clustering. If some bin l contained samples only from P, then the mass in that particular

GPT-2 Size	**Decoding**	**Mauve** (↑)		**Empty bins**	
		No Smoothing	K-T Smoothing	Total Number	Percentage
small	Sampling	$0.589_{0.018}$	$0.655_{0.018}$	$54.2_{6.6}$	$5.4_{0.7}$
	Greedy	0.008	$0.019_{0.000}$	373.0	37.3
	Nucleus, 0.9	$0.878_{0.006}$	$0.906_{0.005}$	$36.4_{4.9}$	$3.6_{0.5}$
medium	Sampling	$0.373_{0.010}$	$0.446_{0.010}$	$77.0_{5.5}$	$7.7_{0.5}$
	Greedy	0.012	0.024	314.0	31.4
	Nucleus, 0.9	$0.915_{0.006}$	$0.936_{0.004}$	$29.0_{6.6}$	$2.9_{0.7}$
large	Sampling	$0.845_{0.010}$	$0.878_{0.008}$	$30.2_{1.3}$	$3.0_{0.1}$
	Greedy	$0.012_{0.000}$	$0.026_{0.000}$	$311.4_{0.8}$	$31.1_{0.1}$
	Nucleus, 0.95	$0.936_{0.003}$	$0.952_{0.002}$	$26.6_{3.0}$	$2.7_{0.3}$
xl	Sampling	$0.882_{0.006}$	$0.908_{0.005}$	$27.6_{6.8}$	$2.8_{0.7}$
	Greedy	0.016	0.033	288.0	28.8
	Nucleus, 0.95	$\mathbf{0.940}_{0.006}$	$\mathbf{0.955}_{0.004}$	$23.4_{2.9}$	$2.3_{0.3}$

Table 8: Comparison of Mauve with vector quantization without any smoothing (the default of (Pillutla et al., 2021)) and with Krichevsky–Trofimov (K-T) smoothing (the default Mauve* in this work). Their Spearman correlation is **1.00**. The last two columns show the number and fraction of empty bins obtained after vector quantization (without smoothing) across both P and Q for the computation of Mauve(P, Q). The subscript of each column denotes the standard deviation over 5 random seeds.

bin of $\hat{Q}_{\mathcal{S},m}(l)$ would be missing, i.e., $\hat{Q}_{\mathcal{S},m}(l) = 0$. Table 8 shows the number and fraction of empty bins. We observe around 2% to 5% empty bins for nucleus and ancestral sampling. The number of empty bins increases with an increasing gap between the two distributions: greedy decoding has around 30% of the bins empty while the best setting (nucleus sampling with the xl model) only has 2.3% of the bins empty. This motivates the use of smoothed distribution estimators even with data-dependent vector quantization.

7.3.3 Direct Estimation from Model Probabilities

In contrast to these previous estimation methods based on model embeddings, we compute Mauve directly using the model probabilities $Q(\cdot)$. Since the human probabilities $P(\cdot)$ are not available to us, we use the probabilities from GPT-2 xl (without reshaping the model probabilities) as a surrogate P'. Then, using samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \sim P$ and $\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_n \sim Q$, we approximate the coordinates of the KL divergence curve by the Monte Carlo estimates

$$\mathrm{KL}(P \| \lambda P + (1 - \lambda Q)) \approx \frac{1}{n} \sum_{i=1}^{n} \log \frac{P'(\boldsymbol{x}_i)}{\lambda P'(\boldsymbol{x}_i) + (1-\lambda) Q(\boldsymbol{x}_i)} \,,$$

and similarly for $\mathrm{KL}(Q \| \lambda P + (1 - \lambda Q))$.

Results. The results are shown in Figure 10. We observe that this direct estimation can identify the right trend for model size for nucleus sampling, but fails to identify the right trend for ancestral sampling for medium $\prec$ small $\prec$ large (see Table 5). Similarly, it fails to identify the right trends for the decoding algorithm, rating ancestral sampling as better than nucleus sampling.

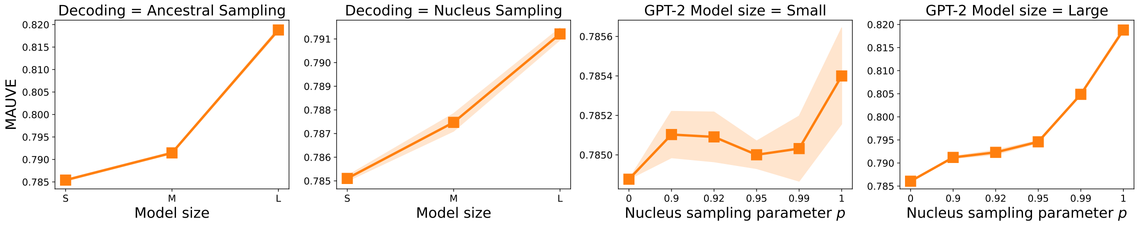


Figure 10: Direct estimation of Mauve from model probabilities $Q(\cdot)$, using the probabilities from GPT-2 xl as a surrogate for the human distribution $P(\cdot)$. The Spearman rank correlation of this direct estimation with Mauve* (the default vector quantization with smoothing) is **0.430** and its worst-case Spearman rank correlation (defined in (30)) with human evaluation scores from Table 5 is **0.371**.

7.3.4 Summary and Discussion

The results of this section show that all the estimation procedures considered in Section 4 can produce useful estimates of the divergence frontier summaries at the right hyperparameter values, while the direct estimation procedure fails. These experiments suggest that the particular vector quantization is not a key factor behind the empirical success of Mauve and refute the argument of Pimentel et al. (2023) that the embedding-based vector quantization is the key ingredient leading to Mauve's strong correlation with human judgments (see §5.1 for a detailed discussion). We note, however, that vector quantization has orthogonal benefits such as its simplicity and fast open-source implementation. As we explore in the upcoming §7.5, a reliable vector embedding turns out to be the key component behind Mauve's strong correlation with human judgment.

7.4 Comparison to Other Divergences and Optimal Transport Costs

Next, we compare our default choice of $\text{Mauve}^\star_{\text{KL}}$ with different f-divergences and optimal transport-based distances.

7.4.1 Divergence Frontier Summaries and Other f-Divergences

We compare Mauve_{KL} with other KL divergence frontier summaries, FI_{KL}, and Mid_{KL}. We also evaluate the corresponding summaries of the χ^2-divergence frontier and two other divergence metrics: the total variation distance $\text{TV}(P, Q)$ and the squared Hellinger distance $H^2(P, Q)$. Since we approximate all the f-divergences in question using vector quantization and Krichevsky–Trofimov (add-1/2) smoothing, we refer to them using their starred names, e.g., $\text{Mauve}^\star_{\text{KL}}$ and $\text{FI}^\star_{\text{KL}}$.

Results. The results are given in Table 9. We see that all divergence frontier summaries correlate perfectly with each other, with a near-perfect Spearman correlation coefficient of 0.99 or higher. Notably, the correlation of FI_{KL} with the Bradley-Terry human evaluation coefficients is larger than the other measures, which are all equal (0.93 versus 0.85 for how human-like the text is). From a closer inspection of the actual values of the various divergences in Table 16 of Appendix D, we see that FI_{KL} ranks ancestral sampling for the xl model as better than nucleus sampling for the small model and agreeing with human

Correlation	$\text{MAUVE}^\star_{\text{KL}}$	$\text{FI}^\star_{\text{KL}}$	$\text{Mid}^\star_{\text{KL}}$	$\text{MAUVE}^\star_{\chi^2}$	$\text{Mid}^\star_{\chi^2}$	$\text{TV}^\star$	$H^2_\star$
$\text{MAUVE}^\star_{\text{KL}}$	1.0	0.99	1.0	1.0	1.0	1.0	1.0
BT/Human-like	0.857	**0.929**	0.857	0.857	0.857	0.857	0.857
BT/Interesting	0.714	**0.738**	0.714	0.714	0.714	0.714	0.714
BT/Sensible	0.762	**0.833**	0.762	0.762	0.762	0.762	0.762

Table 9: Comparison of various divergence frontier summaries and f-divergences with the default $\text{MAUVE}^\star_{\text{KL}}$ and human judgments on the web text dataset. We show their worst-case Spearman rank correlation within one standard deviation (defined in Equation (30)).

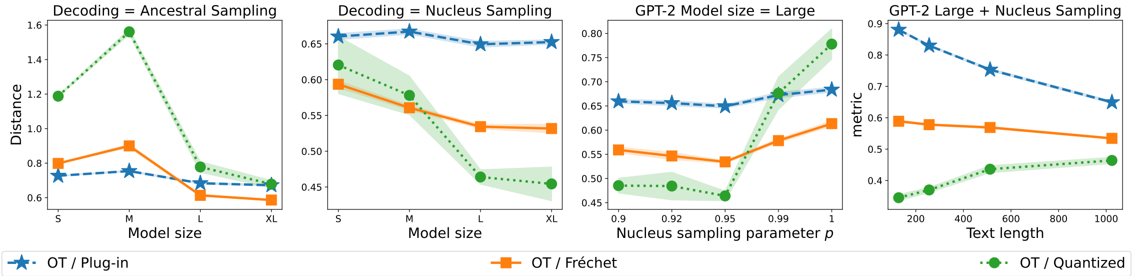


Figure 11: Optimal transport costs for GPT-2 generations in the web text domain. We rescale each measure by a constant so that all the numbers are $O(1)$. Note that a lower transport cost denotes a smaller gap between the distributions. Their correlations with $\text{MAUVE}^\star$ (default) and human evaluations are given in Table 10.

evaluators for how human-like the text is. On the other hand, all other measures (including MAUVE_{KL}) are not able to distinguish between these two in the sense that they are within one standard deviation of each other.

7.4.2 Variants based on Optimal Transport

We investigate divergence frontier summaries based on optimal transport costs rather than f-divergences. Given two distributions $P, Q \in \mathcal{P}(\mathcal{X})$ and a cost function $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, the optimal transport cost between P and Q induced by ρ is defined as

$$\text{OT}_\rho(P, Q) = \min \left\{ \int_{\mathcal{X}\times\mathcal{X}} \rho(\boldsymbol{x}, \boldsymbol{x}')\, \mathrm{d}\pi(\boldsymbol{x}, \boldsymbol{x}') \,:\, \pi \in \mathcal{P}(\mathcal{X} \times \mathcal{X}) \text{ has marginals } P, Q \right\} .$$

In our context, following Section 4.2, we use the cost function

$$\rho(\boldsymbol{x}, \boldsymbol{x}') = \|\varphi(\boldsymbol{x}) - \varphi(\boldsymbol{x}')\|_2^2$$

based on an embedding model $\varphi : \mathcal{X} \to \mathbb{R}^d$. This is also the squared Wasserstein-2 distance between the pushforward distributions $P' = \varphi_\sharp P$ and $Q' = \varphi_\sharp Q$. Similar to Section 4, we simply use the plug-in estimate $\text{OT}_\rho(\hat{P}_n, \hat{Q}_n)$ between the empirical distributions to estimate the optimal transport cost – we refer to it as the **plug-in optimal transport cost**.

We consider quantized versions of this cost following the recipe of Section 4.1. We quantize the empirical distributions $\hat{P}_n$ and $\hat{Q}_n$ into k-dimensional multinomial distributions $\hat{P}_{n,k}, \hat{Q}_{n,k} \in \Delta^{k-1}$. We define a cost $\rho_k(i, j) = \|\boldsymbol{c}_i - \boldsymbol{c}_j\|_2^2$, where $\boldsymbol{c}_i$ is the cluster

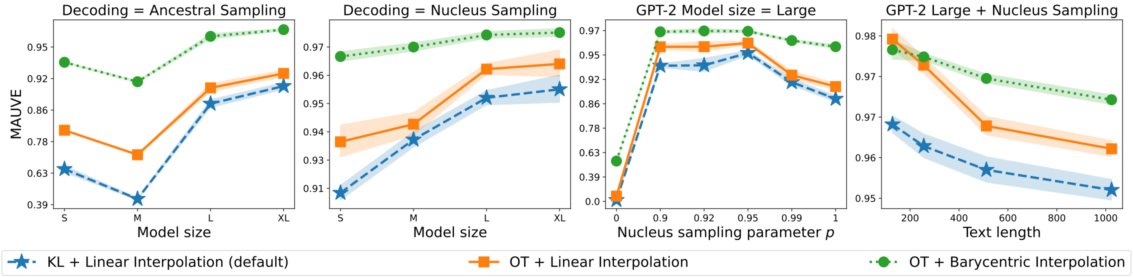


Figure 12: Comparison of variants of MAUVE based on optimal transport costs for GPT-2 generations in the web text domain. Larger values denote a smaller gap for each variant. Their correlations with human evaluations are given in Table 10.

Correlation	OT variants			MAUVE variants		
	Plug-in	Fréchet	Quantized	OT + Linear interpolation	OT + Barycentric interpolation	(Default) KL + Linear interpolation
$\text{MAUVE}^\star_{\text{KL}}$	0.954	0.997	0.980	0.983	0.980	1.000
BT/Human-like	0.810	0.857	0.810	0.810	0.857	0.857
BT/Interesting	0.714	0.714	0.714	0.714	0.714	0.714
BT/Sensible	0.738	0.762	0.738	0.738	0.762	0.762

Table 10: Comparison of optimal transport baselines and variants of MAUVE defined using optimal transport distances with the default $\text{MAUVE}^\star_{\text{KL}}$ and human evaluations on the web text dataset. We show their worst-case Spearman rank correlation within one standard deviation (defined in (30)) for the human evaluations.

center obtained from k-means clustering of the embeddings. We refer to the resulting cost $\text{OT}_{\rho_k}(\hat{P}_{n,k}, \hat{Q}_{n,k})$ as the **quantized optimal transport cost**.

The Fréchet distance (Heusel et al., 2017) is a parametric approximation of OT_ρ which approximates the pushforwards $\varphi_\sharp P$ and $\varphi_\sharp Q$ by multi-variate Gaussians. Note that the approach of Appendix C for MAUVE follows this recipe. Unlike the methods of Appendix C, the Fréchet distance has the advantage that it can be computed in closed form.

We also explore variants of the divergence frontier (Definition 4) based on the optimal transport cost. Define the **optimal transport frontier with linear interpolation** as

$$\mathcal{F}_{\text{OT},\rho}(P,Q) := \left\{ \big(\text{OT}_\rho(P, R_\lambda), \text{OT}_\rho(Q, R_\lambda)\big) \,:\, \lambda \in (0,1) \right\} , \tag{31}$$

where $R_\lambda = \lambda + (1-\lambda)Q$. Inspired by the original characterization of the KL-divergence frontiers as Pareto frontiers (Djolonga et al., 2020), we define a Pareto frontier of optimal transport costs. Concretely, we define the **optimal transport frontier with barycentric interpolation** as

$$\begin{aligned} \mathcal{F}^{\text{bary}}_{\text{OT},\rho}(P,Q) &:= \left\{ \big(\text{OT}_\rho(P, R^\star_\lambda), \text{OT}_\rho(Q, R^\star_\lambda)\big) \,:\, \lambda \in (0,1) \right\} , \\ \text{where} \quad R^\star_\lambda &= \arg\min_R \left\{ \lambda\, \text{OT}_\rho(P,R) + (1-\lambda)\text{OT}_\rho(Q,R) \right\} \end{aligned} \tag{32}$$

is the barycenter of P and Q with weights λ and $1-\lambda$. While the two formulations are equivalent for the KL divergence as we show in Property 3, they are distinct in general for optimal transport costs. The definition in (32) is the analogue of (3) for the KL divergence frontier. We define the corresponding versions of Mauve, namely Mauve_{OT} and $\text{Mauve}_{\text{OT}}^{\text{bary}}$ to be the area under the negative exponential of the frontiers, as in (5).

Computation and Hyperparameter Tuning. Similar to Section 4.1, we estimate the divergence frontiers $\mathcal{F}_{\text{OT}}(P, Q)$ and $\mathcal{F}_{\text{OT}}^{\text{bary}}(P, Q)$ on quantized versions $\mathcal{F}_{\text{OT},\rho_k}(\hat{P}_{n,k}, \hat{Q}_{n,k})$ and $\mathcal{F}_{\text{OT},\rho_k}^{\text{bary}}(\hat{P}_{n,k}, \hat{Q}_{n,k})$. To compute them efficiently, a widely used approach is to add entropic regularization to the optimal transport problem (Cuturi, 2013). Their behavior depends crucially on the regularization parameter being chosen. A good default choice is the median of all the pairwise costs.

Results. The results are shown in Figures 11 and 12, and Table 10.

First, we note that the plug-in optimal transport cost fails to capture the correct dependence for the model size as it rates the medium-sized model as worse than GPT-2 small under nucleus sampling (1499 ± 5 vs. 1473 ± 4, cf. Table 17 in Appendix D). The plug-in estimator also fails to capture the dependence on the text length. Similar to the Fréchet distance in Section 7.2, its numbers suggest that longer model generations drift closer to the human distribution rather than farther away.

This issue of optimal transport costs can be fixed by vector quantization. Indeed, both the quantized optimal transport costs and their frontier summary variants capture the correct dependence in terms of text length, while simultaneously capturing the right trends for the model size and decoding algorithm. This suggests that vector quantization may have a regularizing effect on the estimation problem — we leave a deeper exploration of this phenomenon for future work.

Correlation Analysis. We see from Table 10 that the plug-in optimal transport cost has a smaller worst-case Spearman correlation of 0.810 with human evaluations. This is smaller than Mauve_{KL}, Fréchet, and $\text{Mauve}_{\text{OT}}^{\text{bary}}$ (0.857) and is on par with Gen. PPL. Comparing the full numbers in Table 17 in Appendix D allows us to find the reason for this discrepancy. The quantized OT cost rates GPT-2 small and medium models with nucleus sampling (resp. 0.083 and 0.077) as better than the large and xl models with ancestral sampling (resp. 0.090 and 0.104; these gaps are larger than the standard deviation of 0.005 across runs). These trends disagree with human evaluations. Mauve_{OT} and $\text{Mauve}_{\text{OT}}^{\text{bary}}$ make the same mistake while $\text{Mauve}_{\text{KL}}^{\star}$ identifies the small model with nucleus sampling as being worse than the large and xl models with ancestral sampling.

Summary and Discussion. Naïve use of optimal transport costs such as the Fréchet distance (parametric Gaussian approximation) or the empirical estimator in the embedding space leads to a failure to capture the right trend with respect to the generation length. This issue is specific to the text setting due to the lack of a temporal dimension for images; indeed, the Fréchet distance is the de facto standard evaluation metric for image generation. Optimal transportation in the quantized embedding space (similar to Section 4.1), as well as frontier summaries that build upon them can overcome this issue.

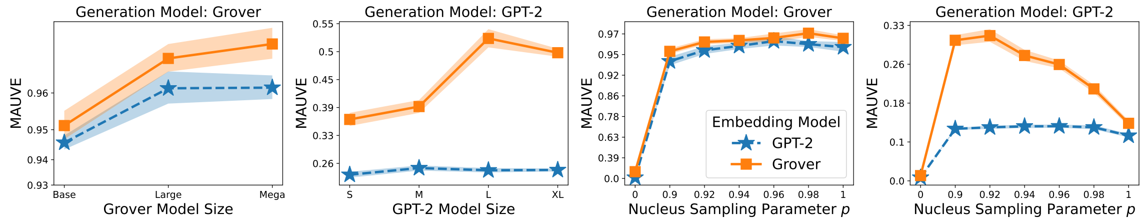


Figure 13: Effect of embeddings on the news generations. We compare generative models GPT-2 and Grover using embeddings from both GPT-2 and Grover. The Spearman rank correlation between $\textsc{Mauve}^\star_{\text{Grover}}(P, \cdot)$ and $\textsc{Mauve}^\star_{\text{GPT-2}}(P, \cdot)$ is **0.971**.

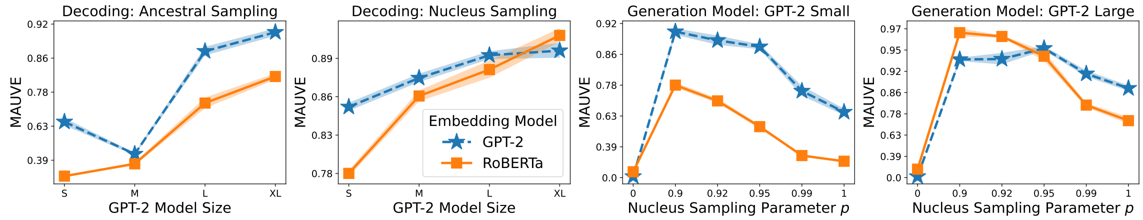


Figure 14: Effect of embeddings on web text generations with GPT-2. $\textsc{Mauve}^\star$ computed from GPT-2 embeddings and RoBERTa embeddings have a Spearman rank correlation of **0.962**.

7.5 Effect of the Embedding

The results of Sections 7.3 and 7.4.1 suggest that the embedding is a key factor in the empirical usefulness of Mauve and other divergence frontier summaries. In this section, we analyze the effect of the embeddings, experimenting with using the generative model itself, using masked language models, shallow embeddings, and finally string-based embeddings that are not learned from data.

7.5.1 Reusing a Generative Model For Embeddings

First, we study whether using the embeddings from the same generative model we are evaluating might bias the proposed measures toward generations from that model. In particular, consider two generative models Q_1 and Q_2, and let $\textsc{Mauve}_i(P, \cdot)$ denote the value of Mauve obtained from using embeddings from model Q_i for $i \in \{1, 2\}$. We check whether $\textsc{Mauve}_1(P, Q_1) > \textsc{Mauve}_1(P, Q_2)$ but $\textsc{Mauve}_2(P, Q_1) < \textsc{Mauve}_2(P, Q_2)$.

We perform a comparison in the news domain, where P denotes the distribution of articles in the RealNews dataset. We take Q_1 to the Grover model and Q_2 to be GPT-2, both of various sizes and decoding algorithms.[9] We use Grover large and GPT-2 large to compute the embeddings.

The results are given in Figure 13. We observe that the embeddings from both GPT-2 and Grover agree that generations from Grover are closer to the RealNews distribution than GPT-2. This trend holds uniformly across model sizes and decoding algorithms. Indeed,

[9] Although the training data of GPT-2 is proprietary, its open version OpenWebText (Gokaslan and Cohen, 2019) contains a significant number of news articles (Sharoff, 2020). The most frequently occurring web domains in OpenWebText are news domains (Gehman et al., 2020, Figure 5).

the Spearman rank correlation between $\textsc{Mauve}_{\text{GPT-2}}$ and $\textsc{Mauve}_{\text{Grover}}$ is **0.971**. Still, there are some minor differences in the trends revealed by each of the features. For instance, Grover embeddings suggest that news generations from GPT-2 large are better than those from GPT-2 xl. Similarly, Grover embeddings suggest $p = 0.92$ as the best nucleus sampling hyperparameter for GPT-2 generations, while features from GPT-2 think $0.9 \le p \le 1$ are roughly equivalent.

Overall, we find that the Mauve scores obtained from both generative models are strongly correlated, and we do not find any evidence of bias from reusing a generative model for embeddings.

7.5.2 Masked Language Model Embeddings

So far, we only considered embeddings from left-to-right language models such as GPT-2 and Grover. In this next experiment, we consider using embeddings from a masked language model, RoBERTa large (Liu et al., 2019). We repeat the experiments in the web text domain with GPT-2 as the generative model and RoBERTa as the embedding model.

The results are given in Figure 14. First, we note that the correlation between Mauve computed from GPT-2 embeddings and RoBERTa embeddings has a Spearman rank correlation of **0.962**. Second, we observe that RoBERTa embeddings also capture the trends concerning model size and decoding, with some minor differences. For instance, both models identify the greedy $\prec$ ancestral $\prec$ nucleus trend from Section 7.2. While both embedding models agree that $p = 0.9$ is the best nucleus sampling hyperparameter for the small model, they disagree on generations from the large model. Other baselines such as Gen. PPL. that do not use embeddings suggest that $p = 0.95$ is the best hyperparameter, agreeing with embeddings from GPT-2. We also note that RoBERTa features do not capture the medium $\prec$ small $\prec$ large $\prec$ xl trend for model sizes under ancestral sampling (cf. Table 5).

In summary, the proposed measures computed with masked language models correlate strongly with those computed from left-to-right language models. They can quantify trends concerning model size and decoding.

7.5.3 Learned Shallow GloVe Embeddings

Next, we examine Mauve equipped with learned embeddings predating the advent of transformer language models. We repeat the web text experiments with GPT-2 generations where Mauve is computed based on the GloVe word embeddings (Pennington et al., 2014).

The GloVe embeddings differ from the deep embeddings of the preceding sections in two ways. First, they are non-contextual, meaning that a word (e.g. “bank”) has the same embedding regardless of the context (e.g. “river *bank*” or the “*Bank* of America”). Second, they are embeddings of whitespace-separated words, as opposed to BPE tokens that are used in transformer language models. Overall, we represent a sequence $\boldsymbol{x} = (w_1, \dots, w_T)$ of words[10] using the average GloVe embedding of words in the vocabulary V_{glove}:

$$\varphi_{\text{glove}}(\boldsymbol{x}) = \frac{1}{T} \sum_{i=1}^{T} \text{GloVe}(w_i) \cdot \mathbb{I}(w_i \in V_{\text{glove}}) \,.$$

[10] We use w_i instead of x_i to emphasize that these are words rather than BPE tokens as in the rest of the paper.

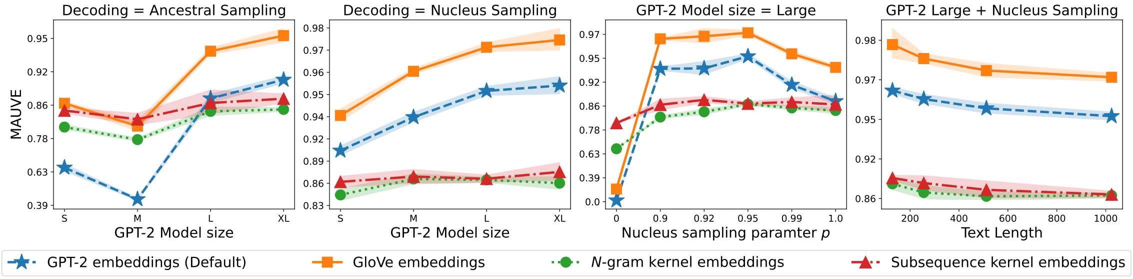


Figure 15: MAUVE from shallow and string-based embeddings on web text generations with GPT-2.

Correlation	GPT-2 Embedding	GloVe Embedding	N-gram Kernel	Subsequence Kernel
$\text{MAUVE}^{\star}_{\text{KL}}$ (default)	1.00	0.993	0.727	0.783
BT/Human-like	0.857	0.928	0.500	0.286
BT/Interesting	0.714	0.738	0.262	0.214
BT/Sensible	0.762	0.833	0.429	0.214

Table 11: Correlation of $\text{MAUVE}^{\star}_{\text{KL}}$ computed from shallow and string-based embeddings with the default GPT-2 embeddings and with human evaluations. For the latter, we show their worst-case Spearman rank correlation within one standard deviation (defined in Equation (30)).

Results. We note that the GloVe embeddings identify the key trends concerning model size, decoding, and text length in Figure 15. Indeed, its worst-case Spearman correlation with the human evaluation in Table 11 is even (marginally) better than that of the GPT-2 embeddings (0.93 vs. 0.86). However, the GloVe embeddings have a significant drawback: they come from a bag-of-words model where word order is irrelevant. As shown in Figure 16, GPT-2 embeddings do not suffer from this drawback. Overall, these results show that MAUVE can extract useful information from shallow GloVe embeddings, demonstrating the versatility of MAUVE.

7.5.4 String-based Kernel Embeddings

Next, we compute MAUVE directly from strings, without any learned embeddings, shallow or deep. Concretely, we consider the embeddings implied by a positive definite kernel $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ between text sequences $\boldsymbol{x}, \boldsymbol{x}'$.

Recall that a kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ over a space $\mathcal{X}$ is said to be positive definite if the Gram matrix $\boldsymbol{K} \in \mathbb{R}^{r \times r}$ with entries $[\boldsymbol{K}]_{i,j} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ defined by any collection $\boldsymbol{x}_1, \dots, \boldsymbol{x}_r \in \mathcal{X}$ of r inputs is a symmetric and positive definite matrix for all integers r; we refer to the textbook (Shawe-Taylor and Cristianini, 2004) for background. A key property of positive definite kernels is that they can be viewed as dot products in an abstract feature space. Specifically, Mercer's theorem states that there is a unique feature map $\varphi_\kappa : \mathcal{X} \to \mathcal{H}$ onto a Hilbert space $\mathcal{H}$ equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ such that $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \langle \varphi_\kappa(\boldsymbol{x}), \varphi_\kappa(\boldsymbol{x}') \rangle_{\mathcal{H}}$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ (Mercer, 1909).

We compute MAUVE using these embeddings φ_κ induced by two string kernels, where $\mathcal{X}$ is the space of text sequences (i.e., strings):

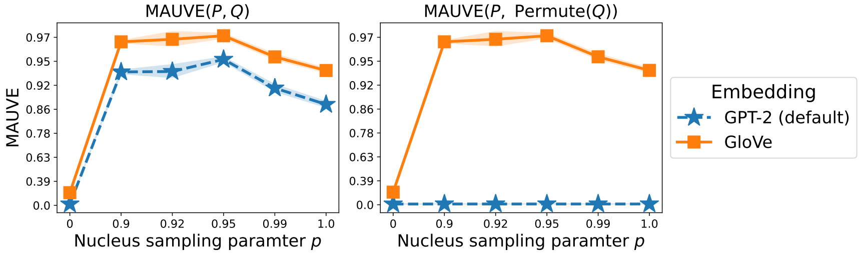


Figure 16: Robustness to permutations at the word level: Mauve with GPT-2 embeddings is sensitive to the order of words whereas GloVe embeddings are not. We define Permute(Q) as the distribution over word sequences $(w_{\pi(1)}, \dots, w_{\pi(n)})$ where $(w_1, \dots, w_n) \sim Q$ and π is a uniformly random permutation on $[n]$.

(a) **N-gram kernel**: Given an integer N, the N-gram kernel is defined as the ratio of common N-grams of its inputs to the total number unique of N-grams (Shawe-Taylor and Cristianini, 2004, Sec. 11.2). Specifically, letting $A_N(\boldsymbol{x})$ denote the set of all N-grams in the sequence $\boldsymbol{x}$, the N-gram kernel κ_N is defined as

$$\kappa_N(\boldsymbol{x}, \boldsymbol{x}') = \frac{|A_N(\boldsymbol{x}) \cap A_N(\boldsymbol{x}')|}{|A_N(\boldsymbol{x}) \cup A_N(\boldsymbol{x}')|} \,.$$

This is also the Jaccard similarity between the set of N-grams of $\boldsymbol{x}$ and those of $\boldsymbol{x}'$.

(b) **Subsequence kernel**: The subsequence kernel (Lodhi et al., 2002) is based on the number of common (non-contiguous) subsequences of length N and scaled by the gap using a decay factor $\lambda \in (0, 1)$, known also as the gap penalty. Concretely, the feature map $\varphi_{N,\lambda}$ used to define the subsequence kernel $\kappa_{N,\lambda}$ has one component for every possible length-N sequence $\boldsymbol{z} \in V^N$. The corresponding component is zero if $\boldsymbol{z}$ is not a subsequence of $\boldsymbol{x}$, else it is

$$\varphi_{N,\lambda}(\boldsymbol{x})[\boldsymbol{z}] = \sum_{\boldsymbol{i}\,:\,\boldsymbol{z}=\boldsymbol{x}[\boldsymbol{i}]} \lambda^{\mathrm{len}(\boldsymbol{i})} \,,$$

where $\boldsymbol{i}$ is an index sequence and $\boldsymbol{x}[\boldsymbol{i}]$ is the subsequence of $\boldsymbol{x}$ obtained by selecting the indices from $\boldsymbol{i}$, and $\mathrm{len}(\boldsymbol{i}) = i_{|\boldsymbol{i}|} - i_1 + 1$ is the length of the subsequence in $\boldsymbol{x}$. A naïve implementation of $\kappa_{N,\lambda}(\boldsymbol{x}, \boldsymbol{x}')$ has a complexity of $O(|V|^N)$ but it can be implemented using sparse dynamic programming in $O(NM \log |\boldsymbol{x}|)$ time, where $M = |\{(i, j) \,:\, x_i = x'_j\}|$ is the total number of matches between $\boldsymbol{x}$ and $\boldsymbol{x}'$ (Rousu et al., 2005).

We compute Mauve from the respective embeddings of these two kernels at the level of word-piece tokens using the nearest neighbor method of §4.2. To keep the Mauve computation time to under two hours, we use $n = 800$ samples for the N-gram kernel and $n = 200$ samples for the subsequence kernel. We sweep over the hyperparameters $N \in \{3, 4, 5\}$ and $\lambda \in \{0.1, 0.2, \dots, 0.9\}$ of the kernels and report the hyperparameters that have the highest correlation with the human evaluation: these are $N = 3$ for the N-gram kernel and $N = 5, \lambda = 0.5$ for the subsequence kernel.

Results. Figure 15 shows the dependence of Mauve on the trends concerning model size, decoding, and text length. We see that string kernel embeddings only identify these trends weakly and unreliably, i.e., the mean across 5 runs trend is as expected but the gaps are often smaller than the standard deviation across runs. This is true of all three trends but take the text length as an example. Mauve from the subsequence kernel at a length of 512 tokens is 0.879 ± 0.013, which is smaller than 0.889 ± 0.010 at length 256 and larger than 0.871 ± 0.005 at 1024 tokens, but all three numbers are within one standard deviation of each other. Similarly, we see from Table 11 that the worst-case Spearman correlations with the human evaluation results are small, always under 0.5. This shows that the raw strings are not informative enough for Mauve.

7.5.5 Summary and Discussion

The results of this subsection demonstrate the importance of the embedding to the usefulness of Mauve. The poor performance of N-gram and subsequence kernels, and direct model probabilities (Section 7.3.3) show that some care must be taken to use informative embeddings. Yet, Mauve is versatile enough to leverage information from a wide variety of embeddings, including language model embeddings (left-to-right LMs, even if it has been used for generation, or masked LMs), and shallow non-contextual embeddings.

7.6 Comparison to Generative Precision and Recall

Metrics based on divergence frontiers have been previously used extensively in the computer vision community (Sajjadi et al., 2018; Kynkäänniemi et al., 2019; Djolonga et al., 2020). How do these metrics fare in the evaluation of text generative models? We now examine the applicability of the most widely used such metrics, i.e., Kynkäänniemi et al.'s precision and recall for generative models, in the web text domain.

Definitions. These notions of precision and recall rely on whether a point $\boldsymbol{x}$ lies within the manifold of a set of samples Y. Concretely, letting $\mathrm{dist}_k(\boldsymbol{z}, Y)$ denoted the distance of $\boldsymbol{z}$ to its k^{th} neighbor in Y, define

$$s_k(\boldsymbol{x}, Y) := \begin{cases} 1, & \text{if } \exists \boldsymbol{z} \in Y \,:\, \rho(\boldsymbol{x}, \boldsymbol{z}) \le \mathrm{dist}_k(\boldsymbol{z}, Y), \\ 0, & \text{else.} \end{cases}$$

Using this notion, the generative precision and recall (evaluated with k nearest neighbors) of a generative distribution Q relative to a target distribution P based on n samples $X_Q \sim Q^n$ and $X_P \sim P^n$ are defined as

$$\mathrm{Precision}_k(X_P, X_Q) = \frac{1}{n} \sum_{\boldsymbol{x}' \in X_Q} s_k(\boldsymbol{x}', X_P), \quad \text{and} \quad \mathrm{Recall}_k(X_P, X_Q) = \frac{1}{n} \sum_{\boldsymbol{x} \in X_P} s_k(\boldsymbol{x}, X_Q) \,.$$

Intuitively, the precision is high if the generated data looks more human-like (i.e., plausibly drawn from P) and the recall is high if the generative model captures the diversity of the target distribution P. Higher values of both precision and recall are desirable. We find that all $1 \le k \le 25$ produced the same qualitative trends, so we show the results for $k = 5$.

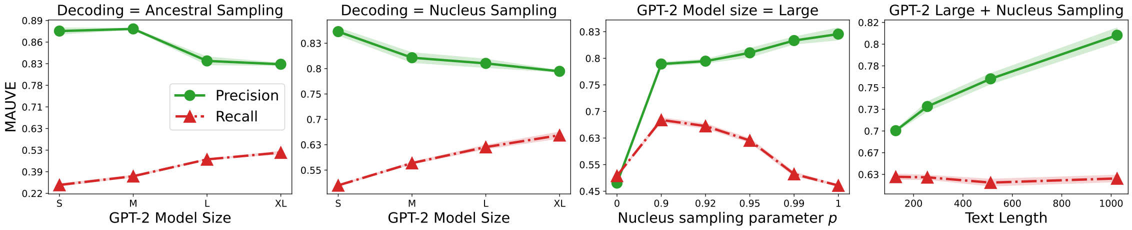


Figure 17: Empirical behavior of generative precision and recall (originally proposed by Kynkäänniemi et al. (2019) for evaluating GANs in computer vision) for natural language generation in the web text domain. Higher values denote better performance.

Results. Figure 17 shows the trends with respect to model scale, decoding algorithm, and text length. Given that larger language models are generally better text generators, we expect the precision and recall to both increase with the model scale. We see for both ancestral and nucleus sampling that the recall increases as expected. However, the precision decreases with increasing model scale; this suggests that smaller models produce more human-like text, which is qualitatively untrue.

Next, we consider the effect of decoding in terms of the nucleus sampling parameter p. Prior work suggests that $p \in [0.9, 0.95]$ should give the most human-like text while $p = 1$ gives the most diverse text (Holtzman et al., 2020). Thus, we would expect the precision to peak in $p \in [0.9, 0.95]$, while we expect the recall to increase with p monotonically. We see that the actual trends are the exact opposite of what we would expect, i.e. $p = 1$ produces the most human-like text whereas $p = 0.9$ best matches the diversity of human text, both of which are qualitatively untrue.

Finally, since model text generations degrade as they get longer, we expect both precision and recall to get worse with text length. Again, the precision metric says that the generated text gets more human-like as its length increases, which is untrue.

Summary and Discussion. In summary, these results demonstrate that the notion of generative precision and recall proposed by Kynkäänniemi et al. do not behave as expected for natural language generation. In contrast, Mauve identifies the expected behavior with respect to model size, decoding algorithm, and text length.

7.7 Evaluating Image Generative Models with Mauve

In this section, we explore the applicability of our approach to measure the gap between a distribution Q of images generated by a neural net and its target distribution P of real-world images.

Setup. We study the distribution of images generated by models trained on the Flickr-Faces-HQ Dataset (FFHQ) (Karras et al., 2019). The models we consider are based on the StyleGAN2-ADA generative adversarial networks described by Karras et al. (2020a).

As a representative divergence frontier summary, we consider $\text{Mauve}^{\star}_{\text{KL}}$ computed using quantization with $k = 1000$ clusters. We use $50,000$ samples from the model in comparison to $50,000$ samples from the FFHQ training data, unless specified otherwise. The resolution of each image is 1024×1024. Note that in the language modality, we compute Mauve

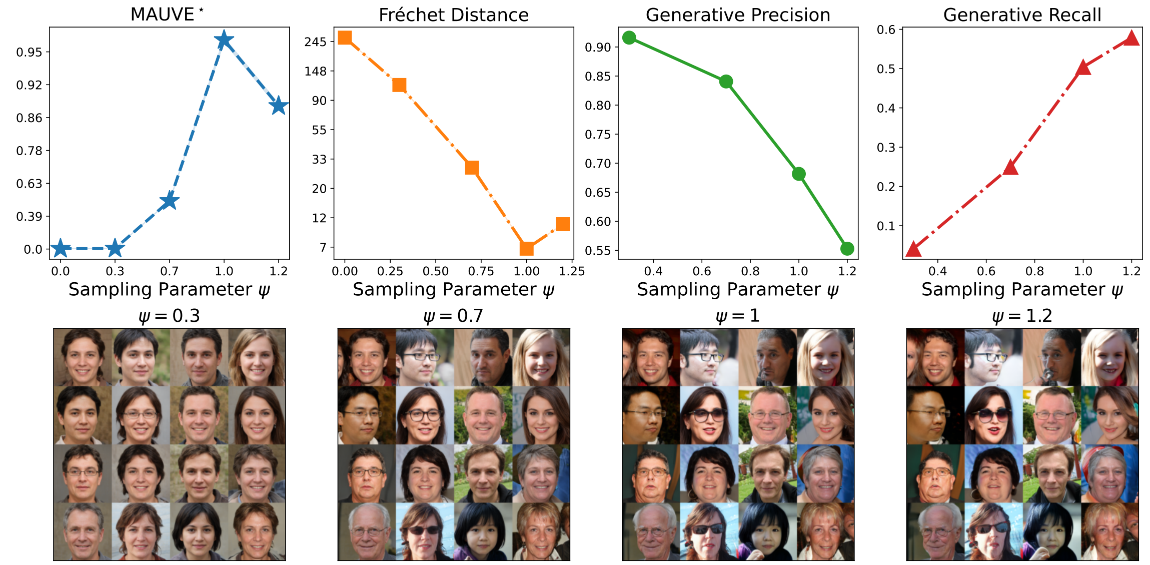


Figure 18: Evaluating image generative models across sampling algorithms with Mauve, Fre´chet distance (Heusel et al., 2017) and generative precision-recall (Kynkäänniemi et al., 2019) (top row). Some sample images generated at various sampling parameters are shown in the bottom row. We use the StyleGAN2-ADA model (Karras et al., 2020a) with various values of the ψ-sampling parameter ψ as the model distribution Q and compare it with the reference distribution P over the FFHQ dataset. The generations shown at each threshold ψ are generated from the same initial randomness for a given position in the grid. We recommend zooming in for a closer inspection of the generated images.

using samples from the *test set* whereas here–following standard practice in vision when comparing distributions using Inception Score or Fréchet distance–we use samples from the *train set*. Similar to these baselines, we use as an embedding model the standard features of an Inception network pre-trained on Imagenet. This setting for Fréchet distance corresponds exactly to the FID-50k metric commonly used in the vision literature. We also compare to the generative precision-recall (Kynkäänniemi et al., 2019); cf. §7.6 for definitions.

7.7.1 Effect of the Sampling

We consider samples drawn from the GAN model using *ψ-sampling*, a technique that biases sampling towards modes of the model distribution.[11]

We briefly describe ψ-sampling. The generator function of these models maps a simple random latent variable $\boldsymbol{z} \sim \mathcal{N}(0, I_{\mathcal{Z}})$ to an image $\boldsymbol{x} = g(\boldsymbol{z}) \in \mathcal{X}$ drawn from the pushforward distribution defined by a learned generator function $g : \mathcal{Z} \to \mathcal{X}$. The generator itself is decomposed into $g = s \circ h$ consisting of an embedding mapping function $h : \mathcal{Z} \to \mathcal{W}$ and synthesis network $s : \mathcal{W} \to \mathcal{X}$. Let $\boldsymbol{w}^* = \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(0, I_{\mathcal{Z}})}[h(\boldsymbol{z})]$ be the average embedding of noise. Given $\boldsymbol{z} \sim \mathcal{N}(0, \mathrm{I}_{\mathcal{Z}})$, we define *$\psi$-sampling* using a modified generator function defined by

$$g_\psi(\boldsymbol{z}) = s(\boldsymbol{w}^* + \psi(h(\boldsymbol{z}) - \boldsymbol{w}^*))\,.$$

[11] ψ-sampling is referred to as *truncation* by Karras et al. (2020a).

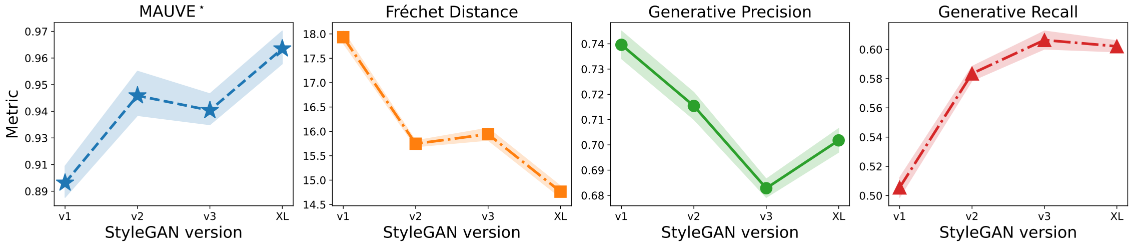


Figure 19: Comparing StyleGAN model generations with Mauve, Frećhet distance (Heusel et al., 2017) and generative precision-recall. We compare: (v1) the original StyleGAN (Karras et al., 2019), (v2) StyleGAN2-ADA (Karras et al., 2020a), (v3) StyleGAN3 (Karras et al., 2021), and (XL) StyleGAN-XL (Sauer et al., 2022). These plots use 5000 samples for each metric, and the shaded region denotes the standard deviation across 10 runs with different subsamples of the target distribution.

If $\psi < 1$, this transformation linearly contracts the mapped value $h(\boldsymbol{z}) \in \mathcal{W}$ towards the mean mapping $\boldsymbol{w}^*$. Intuitively, this will result in higher probability, but less diverse, output images. In contrast, $\psi > 1$ will emphasize the lower probability regions of the image space, resulting in more diverse images of lower quality.

Results. The results are given in Figure 18. Both Mauve and Fréchet distance identify the same ordering of ψ: $1 \succ 1.2 \succ 0.7 \succ 0.3 \succ 0$. Qualitatively, we observe the expected quality-diversity tradeoff as we vary ψ. The extreme $\psi = 0.3$ produces high-quality images of faces that look very similar to each other. At $\psi = 0.7$, we observe more diversity in the generated faces over attributes such as hair color and style, eyewear, and other factors. We get a greater diversity at $\psi = 1$ with more diversity in hair and eyewear but also in the direction the generated face points towards and facial expressions. At $\psi = 1.2$, we that the generated faces start to appear distorted. Some images also feature parts of a second face. The notions of generative precision and recall capture both quality and diversity trends, as was demonstrated in previous work. Thus, similar to Fréchet distance, Mauve accounts for both quality and diversity to produce a single measure of the gap between the model distribution and the target distribution. Unlike both precision-recall and the Fréchet distance, however, Mauve also perfectly identifies various trends in the natural language modality.

7.7.2 Effect of the Model Scale and Architecture

We compare image generative models across different model architectures, analogous to the effect of the model scale in §7.2. For these experiments, we use 5000 samples to compute each metric (compared to the 50000 samples used for the experiments in the previous experiment).

Effect of Model Scale and Architectural Improvements. We compare various generations of StyleGAN models. Each model in this family builds upon the previous one with innovations in the architecture and training pipeline to address certain artifacts in the generated images. We consider the following models:

(a) **StyleGAN** (Karras et al., 2019): the first model in this family with 26 million parameters.

(b) **StyleGAN2-ADA** (Karras et al., 2020b,a): the second model in this family, with 30 million parameters.

(c) **StyleGAN3** (Karras et al., 2021): this model makes substantial changes to the architecture of StyleGAN2. StyleGAN3 has only 15 million parameters but produces image generations of similar quality as StyleGAN2-ADA as per standard metrics like the Fréchet distance. The authors claim that it is better suited to video generation.

(d) **StyleGAN-XL** (Sauer et al., 2022): the largest model in this family that we consider with 71 million parameters.

All images were produced using ψ-sampling with $\psi = 1$.

The results are shown in Figure 19. We find that both Mauve* and Frećhet distance find the same trends: more recent models are better with StyleGAN2-ADA and StyleGAN3 being rated almost the same (i.e., with one standard deviation of each other). Notably, the most recent and the largest model — StyleGAN-XL — produces the best images as per these metrics.

On the other hand, generative precision (Kynkäänniemi et al., 2019) rates the oldest StyleGAN model as producing the most photorealistic images (highest precision). This fails to pass the visual inspection test, as the subsequent works in the StyleGAN family discuss the flaws of this model's generations and are designed to improve them. This is similar to the text domain where generative precision finds that the smallest GPT-2 model produces the most human-like text. Thus, designing fine-grained fidelity and diversity metrics for generative models that can be used reliably across model scales and families remains an important open problem.

Comparing GANs with Diffusion Models. We compare StyleGAN2-ADA with a diffusion model NCSN++ (Song et al., 2021) on the FFHQ domain. NSCN++ is the first diffusion model to directly generate high-resolution images of 1024 × 1024 pixels (without up-sampling lower-resolution images in a multi-step pipeline). Stein et al. (2023) show that diffusion models perform significantly worse than GANs on metrics computed in the Inception-V3 embedding space despite being comparable or better generators in terms of both fidelity (as measured by human evaluations) and diversity. We follow their recommendation and use embeddings from a DINOv2 model (Oquab et al., 2023) (specifically, its ViT-L/14 configuration), which was shown to not have such a bias.

The results are given in Table 12: StyleGAN2-ADA ($\psi = 1$) outperforms the diffusion model by a large margin as per both Mauve and Fréchet distance. Figure 20, which shows some samples from the diffusion model, explains the source of this large disparity. These generations contain more artifacts than the GANs generations (shown in Figure 18), including glaring asymmetries in facial features such as hairs or eyes.

Many successful diffusion-based generative models such as DALL-E 2 (Ramesh et al., 2022) and Imagen (Saharia et al., 2022) adopt a two-step pipeline: (a) generate low-resolution images with a diffusion model (e.g. 64 × 64 pixels), and (b) upsample the generation using one or more super-resolution models (e.g. $64^2 \to 256^2 \to 1024^2$ pixels). Our results above show that end-to-end diffusion modeling to directly generate high-resolution images remains an important open problem.

Model	Fréchet Distance	$\text{Mauve}^{\star}_{\text{KL}}$
StyleGAN2	$298.59_{2.02}$	$0.979_{0.034}$
Diffusion NCSN++	$646.49_{5.80}$	$0.648_{0.002}$

Table 12: GANs vs. diffusion models: Comparing StyleGAN2-ADA with the diffusion model NCSN++ (Song et al., 2021). We use features from the DINOv2 (ViT-L/14) model and each metric is computed using 5000 samples. The subscript denotes the standard deviation over 10 runs with different subsamples of the target distribution. Mauve* is computed using vector quantization of size $k = 100$.

Figure 20: Samples from the diffusion model NCSN++ (Song et al., 2021).

7.7.3 Summary

These results, together with those from the preceding sections, indicate that the general recipe of approximating gaps between distributions of complex high-dimensional objects using embeddings from a pre-trained deep net using f-divergence frontiers and Mauve is a powerful one.

7.8 Tightness of the Statistical Error Bounds

We conduct a numerical study to empirically investigate the tightness of the statistical error bounds presented in Theorem 10. Using the frontier integral FI_{KL} as a representative summary of the f-divergence frontier, we investigate the estimation error in divergence frontier summaries as a function of the sample size n and the quantization size k from samples.

We consider two domains: text generation in the web text domain using a pretrained GPT-2 large and nucleus sampling with $p = 0.95$ (§6) and face image generation using a StyleGAN2-ADA model pretrained on FFHQ sampled using $\psi = 1$ (§7.7).

We study the statistical error incurred by the plug-in estimator using n samples to estimate the population divergence, where each population distribution contains N texts/images ($N = 5000$ for the text domain and $N = 50000$ for the image domain). Following the recipe of §4.1, we first represent each text/image by its features. Next, we quantize these $2N$ features into k bins using k-means clustering. For each support size k, this gives us quantized distributions $P_{\mathcal{S}_k}$ and $Q_{\mathcal{S}_k}$. Then, we sample n i.i.d. examples from each of the two distributions and use their empirical versions $\hat{P}_{\mathcal{S}_k,n}$ and $\hat{Q}_{\mathcal{S}_k,n}$ to compute $\text{FI}(\hat{P}_{\mathcal{S}_k,n}, \hat{Q}_{\mathcal{S}_k,n})$. We estimate the statistical error $\mathbb{E}|\text{FI}_{\text{KL}}(\hat{P}_{\mathcal{S}_k,n}, \hat{Q}_{\mathcal{S}_k,n}) - \text{FI}_{\text{KL}}(P_{\mathcal{S}_k}, Q_{\mathcal{S}_k})|$ from a **Monte Carlo** estimate using 100 random trials and compare it with two bounds from Theorem 10:

(a) **Bound**: the distribution independent bound $(\sqrt{k/n} + k/n)\log n$, and

(b) **Oracle Bound**: the distribution dependent bound $(\alpha_n(P) + \alpha_n(Q))\log n + \beta_n(P) + \beta_n(Q)$ assuming the quantities α_n and β_n (defined in Theorem 10) are known.

We fix the support size (i.e., the quantization size) k and plot each of these quantities in a log-log plot with varying n and compare their *slope*.[12] We then repeat the experiment with

[12] A log-log plot of the function $f(x) = cx^{\lambda}$ is a straight line with slope λ, which thus captures the *degree*.

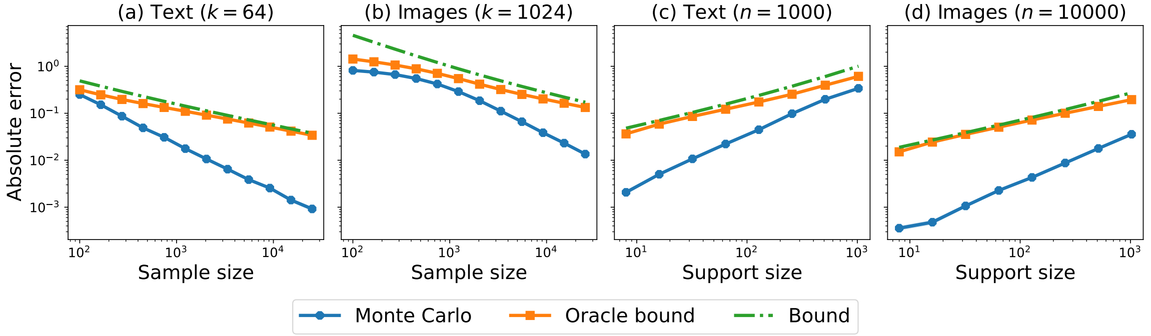


Figure 21: Statistical error of the estimated frontier integral $\mathrm{FI}_{\mathrm{KL}}$ on real text/image data, as a function of the sample size n and support size k. **(a)**: Text data with $k = 64$; **(b)**: Image data with $k = 1024$; **(c)**: Text data with $n = 10^3$; **(d)**: Image data with $n = 10^4$. These bounds are scaled by 30.

n fixed and k varying. We scale the bounds by a factor of 30 for easier visual comparison of their slopes; this only changes the intercept and leaves the slope unchanged.

Results. Figure 21 contains the Monte Carlo estimate and the bounds of the statistical error for real text and image data. In Figure 21(b), we see that the oracle bound captures the right rate for small sample sizes where $k/n > 1$. Whereas, for large n, the distribution-independent bound is better at matching the slope of the Monte Carlo estimate. The same is true for Figure 21(c), where the oracle bound is better for large k. For parts (a) and (d), however, both bounds do not capture the right slope of the Monte Carlo estimate; Theorem 10 is not a tight upper bound in this case. Yet, we notice that Theorem 12 is still a valid upper bound. Indeed, for part (a), we observe that the rate of decrease of the Monte Carlo estimate is only faster than the bound but not slower. Overall, these results demonstrate the favorable statistical error properties of Mauve.

8. Empirical Recommendations

Following the introduction of Mauve in the conference paper (Pillutla et al., 2021), it has been adopted by the language modeling community for measuring performance and hyperparameter tuning in diverse language generation settings, including contrastive decoding (Su et al., 2022; Li et al., 2023), truncation decoding (Meister et al., 2022; Hewitt et al., 2022), and momentum decoding (Lan et al., 2022); controllable text generation (Yang et al., 2023); architectural innovations (Hu et al., 2022); and differentially private language generation (Mattern et al., 2022; Yue et al., 2023; Kurakin et al., 2023).

We review some subtleties of using the proposed measures in practice and offer some practical guidelines.

Aligning Automatic Evaluation to the Goal of Generative Modeling. A common objective of generative modeling is to exactly match the model distribution Q to a real data distribution P. As discussed in Section 3, this can fail due to a Type I error, where the model produces unrealistic or low-quality data, or a Type II error, where the model is unable to produce some plausible real samples and fails to capture the diversity of real data. On the other hand, there are scenarios where ensuring a low Type I error is the only objective of generation (and matching the target P is not important). For instance, correctness is the key objective in machine translation, while using a diverse vocabulary is not the main concern.

The proposed divergence frontier summaries, as measures of the gap between a model distribution Q and real data distribution P, are *well-suited for the first objective and are ill-suited for the second one.* For instance, in the context of open-ended text generation, (Su and Xu, 2022) empirically show that contrastive search has a lower Mauve score than nucleus sampling while producing higher quality text (i.e., lower type I error) as inferred from human evaluations. This can be explained by a large type II error in contrastive search, leading to a large gap or smaller Mauve score. Indeed, each token in contrastive search is chosen deterministically from its top-K vocabulary for small $K < 10$, so it can fail to generate the occasional surprising or low-probability words found in human text (Holtzman et al., 2020).

In summary, we recommend the use of the proposed divergence frontier summaries when the goal of the generative model is to match *both* the quality and diversity of the target real data distribution.

Relative Comparisons Instead of Absolute Scores. We find that the proposed methods are best suited for relative comparisons while the absolute scores are less meaningful. For instance, if we wish to find which model distribution among Q_1 and Q_2 has a smaller gap to the target distribution P, we can compare $\mathrm{Mauve}(P, Q_1)$ to $\mathrm{Mauve}(P, Q_2)$. The individual value of $\mathrm{Mauve}(P, Q_i)$ can vary based on the computational approximation, its hyperparameters, and the number of samples. Indeed, we only consider the rankings induced by Mauve in Section 7 by comparing the Spearman rank correlation with other rankings.

Randomness and Standard Deviations. There are multiple sources of randomness in the computation of Mauve: the randomness from sampling for stochastic decoding algorithms, as well as the random initialization for k-means quantization. Since the absolute values of the proposed measures are not meaningful, the standard deviations are equally important in making relative comparisons. We strongly recommend taking into account the standard deviation across multiple runs rather than just the mean even for relative comparisons; the worst-case Spearman rank correlation defined in (30) is one such measure. We also observed that, while the proposed measures can capture the basic properties as in Section 7.2, it is much harder to quantify subtle differences (e.g., when trying to improve over nucleus sampling). In this case, we recommend increasing the sample size or the number of random seeds to reduce the uncertainty in the statistical estimation.

Sample Size and Text Length. The greater the number of samples, the smaller the statistical estimation error (cf. Section 4). We recommend empirically that each distribution contains at least 1000 samples. The proposed measure computed with a smaller number of samples is biased towards optimism (that is, the score typically goes down as the number of samples increases) and exhibits a larger standard deviation. Likewise, we find that the proposed measures can capture the gap between long texts (at least 256 tokens, preferably 512 tokens) but they might not always capture the difference between shorter texts (see the overlapping shaded areas denoting the standard deviation in Figure 8). In Section 7, we use 5000 samples of up to 1024 tokens (with a prefix length of 35) to compute Mauve and we report the mean and standard deviation over 5 repetitions.

Acknowledgments

Part of this work was done while Zaid Harchaoui was visiting the Simons Institute for the Theory of Computing, and while Krishna Pillutla, Lang Liu, John Thickstun, and Rowan Zellers were at the University of Washington. This work was supported by NSF DMS-2134012, NSF CCF-2019844, NSF DMS-2023166, the DARPA MCS program through NIWC Pacific (N66001-19-2-4031), the CIFAR "Learning in Machines & Brains" program, a Qualcomm Innovation Fellowship, and faculty research awards.

Appendix

The outline of the appendix is as follows:

- Appendix A: Complete proofs of divergence frontier properties from Section 3.
- Appendix B: Full proofs of estimation via quantization from Section 4.1.
- Appendix C: Details of the parametric approximation approach mentioned in Section 4.
- Appendix D: Additional experimental results to augment those in Section 7.
- Appendix E: Additional details of the human evaluations described in Section 6.4.

Appendix A. Properties of the Divergence Frontiers

We give a closed-form expression for Frontier Integral, for the special case of the KL divergence.

Property 18. *The integral summary Frontier Integral of the KL divergence frontier is an f-divergence generated by the convex function*

$$\tilde{f}_{\mathrm{KL}}(t) = \frac{t+1}{2} - \frac{t}{t-1} \log t \,,$$

with the understanding that $\tilde{f}_{\mathrm{KL}}(1) = \lim_{t\to 1} \tilde{f}_{\mathrm{KL}}(t) = 0$*,*

Proof Let P and Q be dominated by some probability measure μ with density p and q, respectively. We will establish the expression

$$\mathrm{FI}(P,Q) = \int_{\mathcal{X}} \mathbb{1}\{p(x) \neq q(x)\} \left(\frac{p(x)+q(x)}{2} - \frac{p(x)q(x)}{p(x)-q(x)} \log \frac{p(x)}{q(x)} \right) \mathrm{d}\mu(x) \,, \tag{33}$$

with the convention $0 \log 0 = 0$. This gives the expression for $\tilde{f}_{\mathrm{KL}}$ from the definition of an f-divergence.

We now establish (33). Denote $\bar\lambda = 1-\lambda$. By Tonelli's theorem, it holds that $\mathrm{FI}_{\mathrm{KL}}(P,Q) = 2\int_{\mathcal{X}} h(p(x), q(x))\mathrm{d}\mu(x)$, where

$$h(p,q) = \int_0^1 \big(\lambda p \log p + \bar\lambda q \log q - (\lambda p + \bar\lambda q)\log(\lambda p + \bar\lambda q)\big)\,\mathrm{d}\lambda.$$

When $p = q$, the integrand is 0. If $q = 0$, then the second term inside the integral is 0, while the first term is $\int_0^1 \lambda p \log(1/\lambda) \mathrm{d}\lambda = p/4$. Finally, when $p \neq q$ are both non-zero, we evaluate the integral to get,

$$h(p, q) = \frac{p}{2} \log p + \frac{q}{2} \log q - \frac{2p^2 \log p - p^2 - 2q^2 \log q + q^2}{4(p - q)} \,,$$

and rearranging the expression gives (33). ■

Next, we give a technical lemma used to establish properties of FI_f and Mid_f.

Proposition 19. *Let $P, Q \in \mathcal{P}(\mathcal{X})$ be probability measures with finite support. Then, the linearized cost $L_{f,\lambda}$ defined in Equation (6) satisfies the bound*

$$L_{f,\lambda}(P\|Q) \le \lambda\, f^*(\lambda) + (1 - \lambda) f^*(1 - \lambda) + 2\lambda(1 - \lambda) f(0) \,.$$

Proof Denote $\bar\lambda = 1 - \lambda$. Let $P, Q \in \Delta^{k-1}$ be discrete distributions over $k < \infty$ items. The function $P, Q \mapsto L_{f,\lambda}(P\|Q)$, by virtue of being an f-divergence, is jointly convex in P, Q. So, $L_{f,\lambda}(P\|Q)$ is maximized for $P^\star, Q^\star$ that lie at some vertices of the probability simplex Δ^{k-1}. We can rule out $P^\star = Q^\star$ as $L_{f,\lambda}(P\|Q) = 0$ in this case. Therefore, without loss of generality, we can assume that $P^\star = (1, 0, \dots, 0) \in \Delta^{k-1}$ and $Q^\star = (0, 1, 0, \dots, 0) \in \Delta^{k-1}$. Plugging this in gives the upper bound

$$L_{f,\lambda}(P^\star\|Q^\star) = \lambda^2 f(1/\lambda) + 2\lambda\bar\lambda f(0) + \bar\lambda^2 f(1/\bar\lambda) = \lambda f^*(\lambda) + \bar\lambda f^*(\bar\lambda) + 2\lambda\bar\lambda f(0) \,.$$

■

Appendix B. Proofs of Theoretical Bounds: Quantization

In this section, we give the complete proofs of quantization in Section 4.1. The outline is as follows:

- Appendix B.1: Proof of the statistical error bound for the empirical estimator (Theorem 10).
- Appendix B.2: Proof of the statistical error bound for the add-constant estimator (Theorem 12).
- Appendix B.3: Proof of the quantization error bound (Proposition 13).

B.1 Statistical Error Bound

In this section, we prove Theorem 10.

The proof relies on two key lemmas—the approximate Lipschitz lemma (Lemma 20) and the missing mass lemma (Lemma 22). The argument breaks into two cases in P (and analogously for Q) for each atom $a \in \mathcal{X}$:

(a) $\hat P_{n,a} > 0$: Since $\hat P_n$ is an empirical measure, we have that $\hat P_{n,a} \ge 1/n$. In this case the approximate Lipschitz lemma gives us the Lipschitzness in $\|P - \hat P_n\|_{\mathrm{TV}}$ up to a factor of $\log n$.

(b) $\hat{P}_{n,a} = 0$: In this case, the mass corresponding to P_a is missing in the empirical measure and we directly bound its expectation following similar arguments as in the missing mass literature; see, e.g., (Berend and Kontorovich, 2012; Mcallester and Ortiz, 2003).

B.1.1 Approximate Lipschitz Property

First, we express the derivatives of $\psi(p,q) = qf(p/q)$ in terms of the derivatives of f:

$$\frac{\partial \psi}{\partial p}(p,q) = f'\left(\frac{p}{q}\right) = f^*\left(\frac{q}{p}\right) - \frac{q}{p}(f^*)'\left(\frac{q}{p}\right) \tag{34a}$$

$$\frac{\partial \psi}{\partial q}(p,q) = f\left(\frac{p}{q}\right) - \frac{p}{q}f'\left(\frac{p}{q}\right) = (f^*)'\left(\frac{q}{p}\right) \tag{34b}$$

$$\frac{\partial^2 \psi}{\partial p^2}(p,q) = \frac{1}{q}f''\left(\frac{p}{q}\right) = \frac{q^2}{p^3}(f^*)''\left(\frac{q}{p}\right) \ge 0 \tag{34c}$$

$$\frac{\partial^2 \psi}{\partial q^2}(p,q) = \frac{p^2}{q^3}f''\left(\frac{p}{q}\right) = \frac{1}{p}(f^*)''\left(\frac{q}{p}\right) \ge 0 \tag{34d}$$

$$\frac{\partial^2 \psi}{\partial p \partial q}(p,q) = -\frac{p}{q^2}f''\left(\frac{p}{q}\right) = -\frac{q}{p^2}(f^*)''\left(\frac{q}{p}\right) \le 0\,, \tag{34e}$$

where the inequalities $f'', (f^*)'' \ge 0$ followed from convexity of f and f^* respectively.

We now present the main lemma that shows that the function ψ is nearly Lipschitz, up to a log factor. This lemma can be leveraged to directly obtain a bound on the statistical error of the f-divergence in terms of the expected total variation distance, provided the probabilities are not too small.

Lemma 20. *Suppose that f satisfies Assumption 9. Consider $\psi : [0,1] \times [0,1] \to [0,\infty)$ given by $\psi(p,q) = qf(p/q)$. We have, for all $p, p', q, q' \in [0,1]$ with $p \vee p' > 0$, $q \vee q' > 0$, that*

$$|\psi(p',q) - \psi(p,q)| \le \left(C_1 \max\left\{1, \log\frac{1}{p \vee p'}\right\} + C_0^* \vee C_2\right)|p - p'|$$
$$|\psi(p,q') - \psi(p,q)| \le \left(C_1^* \max\left\{1, \log\frac{1}{q \vee q'}\right\} + C_0 \vee C_2^*\right)|q - q'|\,.$$

Proof We only prove the first inequality. The second one is identical with the use of f^* rather than f. Suppose $p' \ge p$. From the fact that ψ is convex in p together with a Taylor expansion of $\psi(\cdot, q)$ around p', we get,

$$\begin{aligned}0 \le \psi(p,q) - \psi(p',q) - (p-p')\frac{\partial \psi}{\partial p}(p',q) &= \frac{1}{2}\int_{p'}^{p} \frac{\partial^2 \psi}{\partial p^2}(s,q)(p-s)\mathrm{d}s \\ &= -\frac{p}{2}\int_{p}^{p'} \frac{\partial^2 \psi}{\partial p^2}(s,q)\mathrm{d}s + \frac{1}{2}\int_{p}^{p'} s\frac{\partial^2 \psi}{\partial p^2}(s,q)\mathrm{d}s \\ &\le 0 + C_2(p'-p)\,,\end{aligned}$$

where we used $\partial^2\psi/\partial p^2$ is non-negative due to convexity and, by (34c) and Assumption **(A3)**,

$$s\frac{\partial^2 \psi}{\partial p^2}(s,q) = \frac{s}{q}f''(s/q) \le 2C_2\,.$$

This yields

$$-(p'-p)\frac{\partial\psi}{\partial p}(p',q) \le \psi(p,q) - \psi(p',q) \le -(p'-p)\frac{\partial\psi}{\partial p}(p',q) + C_2(p'-p)\,.$$

We consider two cases based on the sign of $\frac{\partial\psi}{\partial p}(p',q) = f'(p/q)$ (cf. Eq. (34a)).

Case 1. $\frac{\partial\psi}{\partial p}(p',q) \ge 0$. Since $q \mapsto f'(p/q)$ is decreasing in q, we have

$$0 \le (p'-p)\frac{\partial\psi}{\partial p}(p',q) = (p'-p)f'(p/q) \le \lim_{q\to 0}(p'-p)f'(p/q) = (p'-p)f^*(0)\,,$$

where we used $f'(\infty) = f^*(0)$ from Lemma 26. From Assumption **(A1)**, we get the bound

$$|\psi(p,q) - \psi(p',q)| \le (C_0^* \vee C_2)(p'-p)\,.$$

Case 2. $\frac{\partial\psi}{\partial p}(p',q) < 0$. By Assumption **(A2)**, it holds that

$$\left|\frac{\partial\psi}{\partial p}(p',q)\right| \le C_1\,\max\{1, \log(q/p')\} \le C_1\,\max\{1, \log(1/p')\}\,,$$

and thus

$$|\psi(p,q) - \psi(p',q)| \le \left(C_1 \max\left\{1, \log\frac{1}{p'}\right\} + C_2\right)(p'-p)\,.$$

■

With the above lemma, the estimation error of the empirical f-divergence can be upper bounded by the total variation distance between the empirical measure and its population counterpart up to a logarithmic factor, where:

$$\|\hat{P}_n - P\|_{\mathrm{TV}} = \sum_{a\in\mathcal{X}} |\hat{P}_{n,a} - P_a|\,. \tag{35}$$

For the first part, we further upper bound the expected total variation distance of the plug-in estimator, which is

$$\|\hat{P}_n - P\|_{\mathrm{TV}} = \sum_{a\in\mathcal{X}} |\hat{P}_{n,a} - P_a|\,.$$

Lemma 21. *Assume that P is discrete. For any $n \ge 1$, it holds that*

$$\mathbb{E}\|\hat{P}_n - P\|_{\mathrm{TV}} \le \alpha_n(P).$$

Furthermore, if $k = |\mathrm{Supp}(P)| < \infty$, then

$$\mathbb{E}\|\hat{P}_n - P\|_{\mathrm{TV}} \le \alpha_n(P) \le \sqrt{\frac{k}{n}}\,.$$

Proof Using Jensen's inequality, we have,

$$\begin{aligned}\mathbb{E} \sum_{a \in \mathrm{Supp}(P)} |\hat{P}_{n,a} - P_a| &\le \sum_{a \in \mathrm{Supp}(P)} \sqrt{\mathbb{E}(\hat{P}_{n,a} - P_a)^2} \\ &= \sum_{a \in \mathrm{Supp}(P)} \sqrt{\frac{P_a(1-P_a)}{n}} \le \alpha_n(P)\,,\end{aligned}$$

If $k < \infty$, then it follows from Jensen's inequality applied to the concave function $t \mapsto \sqrt{t}$ that

$$\frac{1}{k}\sum_{i=1}^{k} \sqrt{a_k} \le \sqrt{\frac{1}{k}\sum_{i=1}^{k} a_k}\,.$$

Hence, $\alpha_n(P) \le \sqrt{k/n}$ and it completes the proof. ■

B.1.2 Missing Mass Computation

For the second part, we treat the missing mass directly.

Lemma 22 (Missing Mass)**.** *Assume that $k = |\mathrm{Supp}(P)| < \infty$. Then, for any $n \ge 3$,*

$$\mathbb{E}\left[\sum_{a \in \mathcal{X}} \mathbb{1}\{\hat{P}_{n,a} = 0\} P_a\right] \le \frac{k}{n} \tag{36}$$

$$\beta_n(P) := \mathbb{E}\left[\sum_{a \in \mathcal{X}} \mathbb{1}\{\hat{P}_{n,a} = 0\} P_a \left(1 \vee \log\frac{1}{P_a}\right)\right] \le \frac{k \log n}{n}\,, \tag{37}$$

where $a \vee b := \max\{a, b\}$.

Proof We prove the second inequality. The first one is identical. Note that $\mathbb{E}[\mathbb{1}\{\hat{P}_{n,a} = 0\}] = \mathbb{P}(\hat{P}_{n,a} = 0) = (1 - P_a)^n$. Therefore, the left-hand side (LHS) of the second inequality is

$$\begin{aligned}\mathrm{LHS} &= \sum_{a \in \mathcal{X}} (1 - P_a)^n P_a \max\{1, -\log P_a\} \\ &\le \sum_{a \in \mathcal{X}} \frac{1}{n} \vee \frac{\log n}{n} = \frac{k \log n}{n}\,,\end{aligned}$$

where we used Lemma 28 and Lemma 29. ■

Remark 23. *According to (Berend and Kontorovich, 2012, Prop. 3), the bound k/n in (36) is tight up to a constant factor.*

B.1.3 Full Proof of the Statistical bound

Now, we are ready to prove Theorem 10.

Proof [Proof of Theorem 10] Define $\Delta_{n,m}(a) := \left|\psi(P_a, Q_a) - \psi(\hat P_{n,a}, \hat Q_{m,a})\right|$. We have from the triangle inequality that

$$\Delta_{n,m}(a) \le \underbrace{\left|\psi(P_a, Q_a) - \psi(\hat P_{n,a}, Q_a)\right|}_{=:\mathcal{T}_1(a)} + \underbrace{\left|\psi(\hat P_{n,a}, Q_a) - \psi(\hat P_{n,a}, \hat Q_{m,a})\right|}_{=:\mathcal{T}_2(a)} .$$

Since $\hat P_{n,a} = 0$ or $\hat P_{n,a} \ge 1/n$, the approximate Lipschitz lemma (Lemma 20) gives

$$\mathcal{T}_1(a) \le \begin{cases} P_a\left(C_1 \max\{1, \log(1/P_a)\} + C_0^* \vee C_2\right), & \text{if } \hat P_{n,a} = 0, \\ |P_a - \hat P_{n,a}|\left(C_1 \log n + C_0^* \vee C_2\right), & \text{else.} \end{cases}$$

Consequently, Lemma 21 yields

$$\begin{aligned}\sum_{a\in\mathcal{X}} \mathbb{E}[\mathcal{T}_1] &\le \sum_{a\in\mathcal{X}} \mathbb{E}\left[\mathbb{1}\{\hat P_{n,a} = 0\} P_a\left(C_1 \max\{1, \log(1/P_a)\} + C_0^* \vee C_2\right)\right] \\ &\quad + \sum_{a\in\mathcal{X}} \mathbb{E}\left[|\hat P_{n,a} - P_a|\right]\left(C_1 \log n + C_0^* \vee C_2\right) \\ &\le \left(C_1 + C_0^* \vee C_2\right)\beta_n(P) + \big(C_1 \log n + C_0^* \vee C_2\big)\alpha_n(P) .\end{aligned}$$

Since $\psi(p,q) = qf(p/q) = pf^*(q/p)$, an analogous bound holds for $\mathcal{T}_2$ with the appropriate adjustment of constants. Hence, the inequality (10) holds. Moreover, when $k < \infty$, the inequality (11) follows by invoking again Lemma 22 and Lemma 21. ■

We now prove Proposition 11.

Proof [Proof of Proposition 11] The inequality is a direct consequence of Theorem 10. Recall from Property 5 that $D_f(P\|R_\lambda) = D_{f_\lambda}(P\|Q)$ where $f_\lambda(t) := f(t/(\lambda t + 1 - \lambda))(\lambda t + 1 - \lambda)$. From the proof of Theorem 10 we have

$$\begin{aligned}&|D_{f_\lambda}(\hat P_n\|\hat Q_m) - D_{f_\lambda}(P\|Q)| \\ &\le \sum_{a\in\mathcal{X}} \mathbb{1}\{\hat P_{n,a} = 0\}\, P_a\left(C_1 \max\{1, \log(1/P_a)\} + C_0^* \vee C_2\right) \\ &\quad + \sum_{a\in\mathcal{X}} \mathbb{1}\{\hat Q_{m,a} = 0\}\, Q_a\left(C_1^* \max\{1, \log(1/Q_a)\} + C_0 \vee C_2^*\right) \\ &\quad + \sum_{a\in\mathcal{X}} \left|P_a - \hat P_{n,a}\right|\big(C_1 \log n + C_0^* \vee C_2\big) + \sum_{a\in\mathcal{X}} \left|Q_a - \hat Q_{m,a}\right|\big(C_1^* \log m + C_0 \vee C_2^*\big) .\end{aligned}$$

Note that, for the interpolated KL divergence, we have

$$\begin{aligned} C_0 = 1 - \lambda \le 1, &\quad C_0^* = \log\frac{1}{\lambda} - 1 + \lambda \le \log\frac{1}{\lambda_{n,m}} \\ C_1 = 1, &\quad C_1^* = \frac{(1-\lambda)^2}{\lambda} \le \frac{1}{\lambda_{n,m}} \\ C_2 = 1/2, &\quad C_2^* = \frac{1-\lambda}{8\lambda} \le \frac{1}{8\lambda_{n,m}} \end{aligned}$$

for all $\lambda \in [\lambda_{n,m}, 1-\lambda_{n,m}]$. The claim then follows from the same steps of Theorem 10. ■

B.2 Statistical Error Bound with Smoothing

In this section, we apply add-constant smoothing to estimate the f-divergences and study its statistical error.

Consider $P \in \mathcal{P}(\mathcal{X})$ and an i.i.d. sample $\{X_i\}_{i=1}^n \sim P$. The add-constant estimator of P is defined by

$$\hat{P}^b_{n,a} = \frac{N_a + b}{n + kb}, \quad \text{for all } a \in \mathcal{X}\,,$$

where $b > 0$ is a constant and $N_a = |\{i \in [n] : X_i = a\}|$ is the number of times the symbol a appears in the sample. In practice, $b = b_a$ could be different depending on the value of N_a, but we use the same constant b for simplicity. Similarly, We define $\hat{Q}^b_m$ with $M_a = |\{i \in [m] : Y_i = a\}|$. The goal is to upper bound the statistical error

$$\mathbb{E}|D_f(P\|Q) - D_f(\hat{P}^b_n\|\hat{Q}^b_m)| \tag{38}$$

under Assumption 9.

Compared to the statistical error of the plug-in estimator, a key difference is that each entry in the add-constant estimator is at least $(n+kb)^{-1} \wedge (m+kb)^{-1}$. Hence, we can directly apply the approximate Lipschitz lemma without the need to control the missing mass part. Another difference is that the total variation distance is now between the add-constant estimator and its population counterpart, which can be bounded as follows.

Lemma 24. *Assume that $k = \mathrm{Supp}(P) < \infty$. Then, for any $b > 0$,*

$$\sum_{a\in\mathcal{X}} \mathbb{E}|\hat{P}^b_{n,a} - P_a| \le \sum_{a\in\mathcal{X}} \frac{\sqrt{nP_a(1-P_a)} + bk|P_a - 1/k|}{n+kb} \le \frac{\sqrt{kn} + 2b(k-1)}{n+kb}\,.$$

Proof Note that

$$|\hat{P}^b_{n,a} - P_a| = |\frac{N_a - nP_a}{n+kb} + \frac{b(1-kP_a)}{n+kb}| \le |\frac{N_a - nP_a}{n+kb}| + |\frac{b(1-kP_a)}{n+kb}|.$$

Using Jensen's inequality, we have

$$\begin{aligned}\sum_{a\in\mathcal{X}} \mathbb{E}|\hat{P}^b_{n,a} - P_a| &\le \sum_{a\in\mathcal{X}} \left[\sqrt{\mathbb{E}|\frac{N_a - nP_a}{n+kb}|^2} + \frac{c|1-kP_a|}{n+kb}\right] \\ &= \sum_{a\in\mathcal{X}} \left[\frac{\sqrt{nP_a(1-P_a)}}{n+kb} + \frac{bk|1/k - P_a|}{n+kb}\right].\end{aligned}$$

We claim that

$$\sum_{a\in\mathcal{X}} |P_a - \frac{1}{k}| \le \frac{2(k-1)}{k}.$$

If this is true, we have

$$\sum_{a\in\mathcal{X}} \mathbb{E}|\hat{P}^b_{n,a} - P_a| \le \frac{\sqrt{kn} + 2b(k-1)}{n+kb} \,,$$

since $\sum_{a\in\mathcal{X}} \sqrt{P_a(1-P_a)} \le \sqrt{k}$. It then remains to prove the claim. Take $a_1, a_2 \in \mathcal{X}$ such that $P_{a_1} \ge k^{-1} \ge P_{a_2}$. It is clear that

$$\begin{aligned} |P_{a_1} - \frac{1}{k}| + |P_{a_2} - \frac{1}{k}| &\le |P_{a_1} + P_{a_2} - \frac{1}{k}| + |P_{a_2} - P_{a_2} - \frac{1}{k}| \\ &= P_{a_1} + P_{a_2}. \end{aligned}$$

Repeating this argument gives

$$\sum_{a\in\mathcal{X}} |P_a - \frac{1}{k}| \le 1 - \frac{1}{k} + \frac{k-1}{k} = \frac{2(k-1)}{k}.$$

■

Now we are ready to prove Theorem 12.

Proof [Proof of Theorem 12] Following the proof of Theorem 10, we define

$$\Delta_{n,m}(a) := |\psi(P_a, Q_a) - \psi(\hat{P}^b_{n,a}, \hat{Q}^b_{m,a})| \,.$$

We have from the triangle inequality that

$$\Delta_{n,m}(a) \le \underbrace{\left|\psi\big(P_a, Q_a\big) - \psi\big(\hat{P}^b_{n,a}, Q_a\big)\right|}_{=:\mathcal{T}_1(a)} + \underbrace{\left|\psi\big(\hat{P}^b_{n,a}, Q_a\big) - \psi\big(\hat{P}^b_{n,a}, \hat{Q}^b_{m,a}\big)\right|}_{=:\mathcal{T}_2(a)} \,.$$

Since $\hat{P}^b_{n,a} \ge b/(n+kb)$, the approximate Lipschitz lemma (Lemma 20) gives

$$\mathcal{T}_1(a) \le |P_a - \hat{P}^b_{n,a}| \left(C_1 \log(n/b + k) + C_0^* \vee C_2\right),$$

By lemma 24, it holds that

$$\begin{aligned} \frac{\sum_{a\in\mathcal{X}} \mathbb{E}[\mathcal{T}_1(a)]}{C_1 \log(n/b+k) + C_0^* \vee C_2} &\le \sum_{a\in\mathcal{X}} \left[\frac{\sqrt{nP_a}}{n+kb} + \frac{bk|1/k - P_a|}{n+kb}\right] = \frac{n\alpha_n(P)}{n+kb} + \gamma_{n,k}(P) \\ &\le \frac{\sqrt{kn} + 2b(k-1)}{n+kb} \,. \end{aligned}$$

Since $\psi(p,q) = qf(p/q) = pf^*(q/p)$, an analogous bound holds for $\mathcal{T}_2(a)$ with the appropriate adjustment of constants and the sample size. Putting these together, we get,

$$\begin{aligned}
&\mathbb{E}\big|D_f(P\|Q) - D_f(\hat{P}_n^b\|\hat{Q}_m^b)\big| \le \mathbb{E}\left[\sum_{a\in\mathcal{X}} |\Delta_n(a)|\right] \\
&\le \left[\frac{n\alpha_n(P)}{n+kb} + \gamma_{n,k}(P)\right] \big(C_1 \log(n/b+k) + C_0^* \vee C_2\big) \\
&\quad + \left[\frac{m\alpha_m(Q)}{m+kb} + \gamma_{m,k}(Q)\right] \big(C_1^* \log(m/b+k) + C_0 \vee C_2^*\big) \\
&\le \big(C_1 \log(n/b+k) + C_0^* \vee C_2\big)\frac{\sqrt{kn}+2b(k-1)}{n+kb} \\
&\quad + \big(C_1^* \log(m/b+k) + C_0 \vee C_2^*\big)\frac{\sqrt{km}+2b(k-1)}{m+kb}\,.
\end{aligned}$$

■

B.3 Quantization Error

We establish a bound on the quantization error of f-divergences, i.e.,

$$\inf_{|\mathcal{S}|\le k} |D_f(P\|Q) - D_f(P_{\mathcal{S}}\|Q_{\mathcal{S}})|, \tag{39}$$

where the infimum is over all partitions of $\mathcal{X}$ of size no larger than k, and $P_{\mathcal{S}}$ and $Q_{\mathcal{S}}$ are the quantized versions of P and Q according to $\mathcal{S}$, respectively. Note that we do not assume $\mathcal{X}$ to be discrete in this section. All the results hold for the linearized cost $L_\lambda(\hat{P}_n, \hat{Q}_n)$ and the frontier integral $\mathrm{FI}(\hat{P}_n, \hat{Q}_n)$ from Table 1.

Our analysis is inspired by the following result, which shows that the f-divergence can be approximated by its quantized counterpart; see, e.g., (Györfi and Nemetz, 1978, Theorem 6).

Theorem 25. *For any $P, Q \in \mathcal{P}(\mathcal{X})$, it holds that*

$$D_f(P\|Q) = \sup_{\mathcal{S}} D_f(P_{\mathcal{S}}\|Q_{\mathcal{S}}), \tag{40}$$

where the supremum is over all finite partitions of $\mathcal{X}$.

We now prove Proposition 13, the finite-partition analogue of this.

Proof [Proof of Proposition 13] Assume $f(0) + f^*(0) < \infty$. Otherwise, there is nothing to prove. Fix two distributions P, Q over $\mathcal{X}$. Partition the measurable space $\mathcal{X}$ into

$$\mathcal{X}_1 = \left\{x \in \mathcal{X} \,:\, \frac{\mathrm{d}P}{\mathrm{d}Q}(x) \le 1\right\}\,, \quad \text{and,} \quad \mathcal{X}_2 = \left\{x \in \mathcal{X} \,:\, \frac{\mathrm{d}P}{\mathrm{d}Q}(x) > 1\right\}\,,$$

so that

$$D_f(P\|Q) = \int_{\mathcal{X}_1} f\left(\frac{\mathrm{d}P}{\mathrm{d}Q}(x)\right)\mathrm{d}Q(x) + \int_{\mathcal{X}_2} f^*\left(\frac{\mathrm{d}Q}{\mathrm{d}P}(x)\right)\mathrm{d}P(x) =: D_f^+(P\|Q) + D_{f^*}^+(Q\|P)\,.$$

We quantize $\mathcal{X}_1$ and $\mathcal{X}_2$ separately, starting with $\mathcal{X}_1$. Define sets $S_1, \cdots, S_k$ as

$$S_m = \left\{ x \in \mathcal{X}_1 \,:\, \frac{f(0)(m-1)}{k} \le f\left(\frac{\mathrm{d}P}{\mathrm{d}Q}(x)\right) < \frac{f(0)m}{k} \right\},$$

where the last set S_k is also extended to include $\{x \in \mathcal{X}_1 \,:\, f((\mathrm{d}P/\mathrm{d}Q)(x)) = f(0)\}$. Since f is nonincreasing on $(0,1]$, it follows that $\sup_{x\in\mathcal{X}_1} f((\mathrm{d}P/\mathrm{d}Q)(x)) \le f(0)$. As a result, the collection $\mathcal{S} = \{S_1, \cdots, S_k\}$ is a partition of $\mathcal{X}_1$. This gives

$$\frac{f(0)}{k}\sum_{m=1}^{k}(m-1)\,Q[S_m] \le D_f^+(P\|Q) \le \frac{f(0)}{k}\sum_{m=1}^{k} m\,Q[S_m]\,. \tag{41}$$

Further, since f is nonincreasing on $(0,1]$, we also have

$$\frac{f(0)(m-1)}{k} \le f\left(\sup_{x\in F_m}\frac{\mathrm{d}P}{\mathrm{d}Q}(x)\right) \le f\left(\frac{P[F_m]}{Q[F_m]}\right) \le f\left(\inf_{x\in F_m}\frac{\mathrm{d}P}{\mathrm{d}Q}(x)\right) \le \frac{f(0)m}{k}\,.$$

Hence, it follows that

$$\frac{f(0)}{k}\sum_{m=1}^{k}(m-1)\,Q[S_m] \le D_f^+(P_{\mathcal{S}_1}\|Q_{\mathcal{S}_1}) \le \frac{f(0)}{k}\sum_{m=1}^{k} m\,Q[S_m]\,. \tag{42}$$

Putting (41) and (42) together gives

$$\inf_{|\mathcal{S}_1|\le k}\left|D_f^+(P\|Q) - D_f^+(P_{\mathcal{S}_1}\|Q_{\mathcal{S}_1})\right| \le \frac{f(0)}{k}\sum_{m=1}^{k} Q[S_m] \le \frac{f(0)}{k}\,, \tag{43}$$

since $\sum_{m=1}^{k} Q[S_m] = Q[\mathcal{X}_1] \le 1$. Repeating the same argument with P and Q interchanged and replacing f by f^* gives

$$\inf_{|\mathcal{S}_2|\le k}\left|D_{f^*}^+(Q\|P) - D_{f^*}^+(Q_{\mathcal{S}_2}\|P_{\mathcal{S}_2})\right| \le \frac{f^*(0)}{k}\,. \tag{44}$$

To complete the proof, we upper bound the inf of $\mathcal{S}$ over all partitions of $\mathcal{X}$ with $|\mathcal{S}| = k$ by the inf over $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$ with partitions $\mathcal{S}_1$ of $\mathcal{X}_1$ and $\mathcal{S}_2$ of $\mathcal{X}_2$, and $|\mathcal{S}_1| = |\mathcal{S}_2| = k$. Now, under this partitioning, we have, $D_f^+(P_\mathcal{S}\|Q_\mathcal{S}) = D_f^+(P_{\mathcal{S}_1}\|Q_{\mathcal{S}_1})$ and $D_{f^*}^+(Q_\mathcal{S}\|P_\mathcal{S}) = D_{f^*}^+(Q_{\mathcal{S}_2}\|P_{\mathcal{S}_2})$. Putting this together with the triangle inequality, we get,

$$\begin{aligned}
&\inf_{|\mathcal{S}|\le 2k}\left|D_f(P\|Q) - D_f(P_\mathcal{S}\|Q_\mathcal{S})\right| \\
&\le \inf_{\mathcal{S}=\mathcal{S}_1\cup\mathcal{S}_2}\left\{\left|D_f^+(P\|Q) - D_f^+(P_\mathcal{S}\|Q_\mathcal{S})\right| + \left|D_{f^*}^+(Q\|P) - D_{f^*}^+(Q_\mathcal{S}\|P_\mathcal{S})\right|\right\} \\
&\le \inf_{|\mathcal{S}_1|\le k}\left|D_f^+(P\|Q) - D_f^+(P_{\mathcal{S}_1}\|Q_{\mathcal{S}_1})\right| + \inf_{|\mathcal{S}_2|\le k}\left|D_{f^*}^+(Q\|P) - D_{f^*}^+(Q_{\mathcal{S}_2}\|P_{\mathcal{S}_2})\right| \\
&\le \frac{f(0)+f^*(0)}{k}\,.
\end{aligned}$$

■

B.4 Properties and Technical Lemmas

Lemma 26. *Suppose the generator f satisfies Assumptions* ***(A1)*** *and* ***(A2)****. Then,*

$$\lim_{t\to\infty} f'(t) = f^*(0)\,, \quad \text{and} \lim_{t\to\infty} (f^*)'(t) = f(0)\,.$$

Proof We start by observing that

$$\lim_{t\to 0} t|f'(t)| \le C_1 \lim_{t\to 0} t \vee t\log\frac{1}{t} = 0\,.$$

Next, a direct calculation gives

$$(f^*)'(1/t) = f(t) - tf'(t)\,,$$

so that taking the limit $t \to 0$ gives

$$\lim_{t\to\infty} (f^*)'(t) = f(0) - \lim_{t\to 0} tf'(t) = f(0)\,.$$

The proof of the other part is identical. ■

Property 27. *Suppose $f : (0,\infty) \to [0,\infty)$ is convex and continuously differentiable with $f(1) = 0 = f'(1)$. Then, $f'(x) \le 0$ for all $x \in (0,1)$ and $f'(x) \ge 0$ for all $x \in (1,\infty)$.*

Proof Monotonicity of f' means that we have for any $x \in (0,1)$ and $y \in (1,\infty)$ that $f'(x) \le f'(1) = 0 \le f'(y)$. ■

Lemma 28. *For all $x \in (0,1)$ and $n \ge 3$, we have*

$$0 \le (1-x)^n x \log\frac{1}{x} \le \frac{\log n}{n}\,.$$

Proof Let $h(x) = (1-x)^n x\log(1/x)$ be defined on $(0,1)$. Since $\lim_{x\to 0} h(x) = 0 < h(1/n)$, the global supremum does not occur as $x \to 0$. We first argue that h obtains its global maximum in $(0, 1/n]$. We calculate

$$h'(x) = (1-x)^{n-1}\left(-nx\log\frac{1}{x} + (1-x)\left(\log\frac{1}{x} - 1\right)\right) \le (1-x)^{n-1}(1-nx)\log\frac{1}{x}\,.$$

Note that $h'(x) < 0$ for $x > 1/n$, so h is strictly decreasing on $(1/n, 1)$. Therefore, it must obtain its global maximum on $(0, 1/n]$. On this interval, we have,

$$(1-x)^n x\log\frac{1}{x} \le x\log\frac{1}{x} \le \frac{\log n}{n}\,,$$

since $x\log(1/x)$ is increasing on $(0, \exp(-1))$. ■

The next lemma comes from (Berend and Kontorovich, 2012, Theorem 1).

Lemma 29. *For all $x \in (0,1)$ and $n \ge 1$, we have*

$$0 \le (1-x)^n x \le \exp(-1)/(n+1) < 1/n\,.$$

Appendix C. Estimation of Divergences via Parametric Approximations

We discuss here the parametric approximation approach for divergence estimation mentioned in Section 4. Given an embedding model $\varphi : \mathcal{X} \to \mathbb{R}^d$, we first approximate the f-divergence $D_f(P\|Q)$ by $D_f(\varphi_\sharp P\|\varphi_\sharp Q)$. Since $\{\varphi(\boldsymbol{x}_i)\}_{i=1}^n$ is an i.i.d. sample from $\varphi_\sharp P$, we then approximate $\varphi_\sharp P$ by a multivariate Gaussian distribution with mean and covariance matrix given by

$$\hat\mu_P := \frac{1}{n}\sum_{i=1}^n \varphi(\boldsymbol{x}_i) \quad \text{and} \quad \hat\Sigma_P := \frac{1}{n-1}\sum_{i=1}^n (\varphi(\boldsymbol{x}_i) - \hat\mu_P)(\varphi(\boldsymbol{x}_i) - \hat\mu_P)^\top,$$

respectively. The distribution $\varphi_\sharp Q$ can be approximated by $\mathcal{N}_d(\hat\mu_Q, \hat\Sigma_Q)$ similarly. Finally, we approximate $D_f(\varphi_\sharp P\|\varphi_\sharp Q)$ by

$$D_f\left(\mathcal{N}_d(\hat\mu_P, \hat\Sigma_P) \,\|\, \mathcal{N}_d(\hat\mu_Q, \hat\Sigma_Q)\right) = \int f\left(\frac{\phi(\boldsymbol{z}; \hat\mu_P, \hat\Sigma_P)}{\phi(\boldsymbol{z}; \hat\mu_Q, \hat\Sigma_Q)}\right)\phi(\boldsymbol{z}; \hat\mu_Q, \hat\Sigma_Q)\mathrm{d}\boldsymbol{z}\,, \tag{45}$$

where $\phi(\cdot\,; \mu, \Sigma)$ is the probability density function of the multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$. To evaluate the integration in (45), we can use the Monte Carlo approach—(i) generate i.i.d. samples $\{\boldsymbol{z}_b\}_{b=1}^B$ from $\mathcal{N}_d(\hat\mu_Q, \hat\Sigma_Q)$, and (ii) approximate (45) by the empirical average

$$\hat D_f(\mu_P, \Sigma_P\|\mu_Q, \Sigma_Q) = \frac{1}{B}\sum_{b=1}^B f\left(\frac{\phi(\boldsymbol{z}_b; \hat\mu_P, \hat\Sigma_P)}{\phi(\boldsymbol{z}_b; \hat\mu_Q, \hat\Sigma_Q)}\right). \tag{46}$$

Although this approach is widely used in practice, it has no theoretical guarantee. Its performance can get arbitrarily bad depending on the two distributions P and Q. We give below a simple example to illustrate this.

Example 30. *Consider two distributions $\varphi_\sharp P \sim \frac{1}{2}\mathcal{N}(-\mu, 1) + \frac{1}{2}\mathcal{N}(\mu, 1)$ and $\varphi_\sharp Q \sim \mathcal{N}(0, 1)$. It is straightforward to get that $\varphi_\sharp P$ has mean zero and variance*

$$\int x^2 \mathrm{d}\varphi_\sharp P(x) = \frac{1}{2}\int x^2\phi(x; -\mu, 1)\mathrm{d}x + \frac{1}{2}\int x^2\phi(x; \mu, 1)\mathrm{d}x = 1 + \mu^2.$$

As a result, the KL divergence $\mathrm{KL}(\varphi_\sharp P\|\varphi_\sharp Q)$ can be approximated by

$$\mathrm{KL}\left(\mathcal{N}(0, 1+\mu^2)\|\mathcal{N}(0,1)\right) = \frac{\mu^2 - \log(1+\mu^2)}{2}.$$

On the other hand, we also know that

$$\begin{aligned}\mathrm{KL}(\varphi_\sharp P\|\varphi_\sharp Q) &= \int \left[\frac{1}{2}\phi(x; -\mu, 1) + \frac{1}{2}\phi(x; \mu, 1)\right]\log\left[\frac{1}{2}\frac{\phi(x; -\mu, 1)}{\phi(x; 0, 1)} + \frac{1}{2}\frac{\phi(x; \mu, 1)}{\phi(x; 0, 1)}\right]\mathrm{d}x \\ &= \int \phi(x; \mu, 1)\log\left[\frac{1}{2}\frac{\phi(x; -\mu, 1)}{\phi(x; 0, 1)} + \frac{1}{2}\frac{\phi(x; \mu, 1)}{\phi(x; 0, 1)}\right]\mathrm{d}x.\end{aligned}$$

Notice that

$$\frac{1}{2}\frac{\phi(x;-\mu,1)}{\phi(x;0,1)}+\frac{1}{2}\frac{\phi(x;\mu,1)}{\phi(x;0,1)}=\frac{1}{2}e^{-\mu^2/2}\left(e^{-x\mu}+e^{x\mu}\right)=\frac{1}{2}e^{-\mu^2/2}e^{-x\mu}(1+e^{2x\mu}).$$

As a result, we get

$$\begin{aligned}\mathrm{KL}(\varphi_\sharp P\|\varphi_\sharp Q)&=-\frac{\mu^2}{2}-\log 2-\mu\int x\,\phi(x;\mu,1)\,\mathrm{d}x+\int\log\left(1+e^{2x\mu}\right)\phi(x;\mu,1)\,\mathrm{d}x\\&=-\frac{\mu^2}{2}-\log 2-\mu^2+\int\log\left(1+e^{2x\mu}\right)\phi(x;\mu,1)\,\mathrm{d}x\\&\geq-\frac{3\mu^2}{2}-\log 2+\int 2x\mu\,\phi(x;\mu,1)\,\mathrm{d}x\\&=\frac{\mu^2}{2}-\log 2.\end{aligned}$$

This implies that

$$\mathrm{KL}(\varphi_\sharp P\|\varphi_\sharp Q)-\mathrm{KL}\left(\mathcal{N}(0,1+\mu^2)\|\mathcal{N}(0,1)\right)\geq\frac{1}{2}\log\left(1+\mu^2\right)-\log 2$$

can be arbitrarily large as μ increases. Hence, the parametric approximation approach can be extremely inaccurate even in this simple example.

Computational Complexity. Estimating the mean vectors and covariance matrices takes $O(nd^2)$ time. Since evaluating the density $\phi(z;\mu,\Sigma)$ involves computing the quadratic form $(z-\mu)^\top\Sigma^{-1}(z-\mu)$, we can compute Σ^{-1} once with time complexity $O(d^3)$ and evaluate $\Sigma^{-1}(z-\mu)$ for different z's where each evaluation cost $O(d^2)$ time. Assuming that sampling an observation from $\mathcal{N}_d(\hat\mu_Q,\hat\Sigma_Q)$ takes $O(d)$ time, the time complexity of the Monte Carlo approximation is $O(Bd^2+d^3)$. Hence, the parametric approximation approach has overall time complexity $O(nd^2+Bd^2+d^3)$.

Appendix D. Experiments: Additional Results

We elaborate on the results in Section 7 in this section as follows.

- Appendix D.1: full results across model size and decoding for all domains.
- Appendix D.2: full results across text length.
- Appendix D.3: full comparison to other f-divergence frontier summaries.
- Appendix D.4: use of Mauve for hyperparameter tuning.

D.1 Results for Other Domains: News and Stories

The analogue of Table 5 for the news and story domains are Tables 13 and 14 respectively. These are qualitatively similar to the web text domain. Mauve, like discrimination accuracy, rates larger models as better and nucleus sampling as better than ancestral sampling and greedy decoding. An exception to this rule is Grover large, where Mauve thinks ancestral sampling is better than nucleus sampling. The statistics-based measures, namely Zipf coefficient, Repetition, and the fraction of distinct 4 grams all prefer smaller Grover sizes.

Grover Size	Decoding	Gen. PPL	Zipf Coef.	Rep.	Distinct-4	Self-BLEU	% Disc. Acc.(↓)	Mauve*(↑)
base	Sampling	37.505	0.942	0.002	0.882	0.419	99.925	0.754
	Greedy	1.413	1.038	0.518	0.081	0.548	100.000	0.012
	Nucleus, 0.96	23.064	0.974	0.006	**0.847**	0.462	99.950	0.764
large	Sampling	27.796	0.946	**0.002**	0.878	0.429	99.450	0.836
	Greedy	1.575	1.012	0.366	0.124	0.504	100.000	0.013
	Nucleus, 0.98	20.792	**0.962**	0.002	0.859	0.450	98.475	0.800
mega	Sampling	22.656	0.950	0.001	0.879	0.427	97.300	0.847
	Greedy	1.796	1.003	0.316	0.176	0.500	100.000	0.013
	Nucleus, 0.96	**14.834**	0.972	0.003	0.848	**0.469**	**88.675**	**0.852**
Human	n/a	15.356	0.956	0.002	0.842	0.473	–	–

Table 13: News generation evaluation across different Grover model sizes, and decoding approaches. For nucleus sampling, we show the best hyperparameter value from $\{0.9, 0.92, 0.94, 0.96, 0.98\}$ as per Mauve. Disc. Acc. denotes the discrimination accuracy (%) of a Grover large model trained to distinguish human text from machine text generated with the model and decoding algorithm of each row. Boldfaced numbers indicate the performance closest to the human reference when applicable, or the best performance according to the measure.

Decoding	Gen. PPL	Zipf Coef.	REP	Distinct-4	Self-BLEU	% Disc. Acc. (↓)	Mauve*(↑)
Sampling	$38.983_{0.143}$	$\mathbf{1.066}_{0.002}$	$\mathbf{0.001}_{0.000}$	$0.833_{0.001}$	$0.518_{0.003}$	$78.098_{0.365}$	$0.929_{0.007}$
Nucleus, 0.9	$15.433_{0.042}$	$1.201_{0.002}$	$0.006_{0.001}$	$0.719_{0.001}$	$0.637_{0.002}$	$75.150_{0.373}$	$0.914_{0.005}$
Nucleus, 0.92	$17.422_{0.060}$	$1.179_{0.002}$	$0.004_{0.001}$	$0.742_{0.001}$	$0.620_{0.003}$	$71.979_{0.594}$	$0.926_{0.003}$
Nucleus, 0.95	$\mathbf{21.599}_{0.127}$	$1.147_{0.002}$	$0.003_{0.000}$	$\mathbf{0.775}_{0.002}$	$0.589_{0.005}$	$\mathbf{68.586}_{0.583}$	$\mathbf{0.940}_{0.003}$
Top-50	$13.735_{0.027}$	$1.293_{0.004}$	$0.002_{0.000}$	$0.706_{0.001}$	$0.664_{0.003}$	$83.549_{0.381}$	$0.886_{0.010}$
Top-100	$16.527_{0.041}$	$1.252_{0.001}$	$0.002_{0.000}$	$0.743_{0.001}$	$0.631_{0.001}$	$78.150_{0.207}$	$0.913_{0.005}$
Top-500	$23.833_{0.076}$	$1.153_{0.001}$	$0.001_{0.000}$	$0.794_{0.001}$	$\mathbf{0.576}_{0.002}$	$69.680_{0.450}$	$\mathbf{0.942}_{0.004}$
Greedy	1.739	1.362	0.988	0.101	0.742	99.712	0.013
Human	19.704	1.101	0.001	0.783	0.571		

Table 14: Story continuation evaluation across different decoding approaches with GPT-2 medium. Disc. Acc. denotes the discrimination accuracy (%) of a classifier (a frozen GPT-2 large model with a classification head) trained to distinguish human text from machine text generated with the decoding algorithm of each row. Boldfaced numbers indicate the performance closest to the human reference when applicable, or the best performance according to the measure.

Next, we turn to the language modeling comparison measures in Table 15. JS consistently favors greedy decoding, which produces far worse text than other decoding algorithms. Likewise, ε-PPL favors ancestral sampling, which also produces somewhat degenerate text (Holtzman et al., 2020), while SP appears to be unable to distinguish between ancestral sampling and nucleus sampling. This makes SP, JS, and ε-PPL unsuitable to compare generated text to human text.

D.2 Effect of Text Length

We now turn to the plot of comparison measures versus text length in Figure 22. This shows the results of Figure 8 for different hyperparameters. Recall that we expect the quality of the generation to degrade as the maximum length of the text (both machine and human-written) increases.

GPT-2 Size	Decoding	SP(↑)	JS(↓)	ε-PPL(↓)	Human/BT(↑)	Mauve*(↑)
small	Greedy	0.431	0.394	1049.589	–	0.019
	Sampling	0.653	0.425	19.401	−27.52	$0.655_{0.018}$
	Nucleus, 0.9	0.652	0.414	25.938	−15.78	$0.906_{0.005}$
medium	Greedy	0.465	0.371	708.057	–	0.024
	Sampling	0.670	0.402	14.631	−30.77	$0.446_{0.010}$
	Nucleus, 0.9	0.670	0.391	18.821	−3.43	$0.936_{0.004}$
large	Greedy	0.483	0.359	580.020	–	0.026
	Sampling	0.679	0.381	12.658	−6.93	$0.878_{0.008}$
	Nucleus, 0.95	0.679	0.374	14.938	12.55	$0.952_{0.002}$
xl	Greedy	0.496	**0.349**	497.696	–	0.033
	Sampling	**0.686**	0.369	**11.412**	8.97	$0.908_{0.005}$
	Nucleus, 0.95	0.686	0.363	13.677	**15.66**	$\mathbf{0.955}_{0.004}$
	Adversarial	n/a	n/a	n/a	–	0.057

Table 15: Mauve versus comparison measures based on language modeling (SP, JS, and ε-PPL) across different model sizes, and decoding approaches for web text generations. SP, JS, and ε-PPL are deterministic because they do not require generations from a decoding algorithm. Moreover, they cannot measure the quality of the adversarial decoding. The column "Human/BT" gives the Bradley-Terry score obtained from a pairwise human evaluation (Section 7.1). Boldfaced numbers indicate the best performance according to the measure.

GPT-2 Size	Decoding	Mauve$^\star_{\mathrm{KL}}$ (↑)	FI$^\star_{\mathrm{KL}}$ (↓)	Mid$^\star_{\mathrm{KL}}$ (↓)	Mauve$^\star_{\chi^2}$ (↑)	Mid$^\star_{\chi^2}$ (↓)	TV* (↓)	$H^2_\star$ (↓)	BT (↑) Human-like
small	Sampling	$0.655_{0.018}$	$0.033_{0.002}$	$0.105_{0.004}$	$0.335_{0.020}$	$0.191_{0.007}$	$0.363_{0.006}$	$0.225_{0.010}$	−27.518
	Nucleus, 0.9	$0.906_{0.005}$	$0.016_{0.001}$	$0.044_{0.001}$	$0.734_{0.011}$	$0.084_{0.003}$	$0.230_{0.005}$	$0.091_{0.003}$	−15.783
medium	Sampling	$0.446_{0.010}$	$0.042_{0.001}$	$0.160_{0.003}$	$0.164_{0.004}$	$0.277_{0.003}$	$0.443_{0.004}$	$0.356_{0.009}$	−30.769
	Nucleus, 0.9	$0.936_{0.004}$	$0.012_{0.001}$	$0.035_{0.001}$	$0.805_{0.009}$	$0.068_{0.002}$	$0.205_{0.004}$	$0.073_{0.002}$	−3.429
large	Sampling	$0.878_{0.008}$	$0.017_{0.000}$	$0.052_{0.002}$	$0.672_{0.016}$	$0.098_{0.003}$	$0.251_{0.004}$	$0.107_{0.004}$	−6.935
	Nucleus, 0.95	$0.952_{0.002}$	$0.010_{0.000}$	$0.030_{0.001}$	$0.849_{0.007}$	$0.058_{0.002}$	$0.187_{0.005}$	$0.061_{0.002}$	12.553
xl	Sampling	$0.908_{0.005}$	$0.014_{0.001}$	$0.044_{0.001}$	$0.737_{0.012}$	$0.083_{0.003}$	$0.232_{0.005}$	$0.090_{0.003}$	8.966
	Nucleus, 0.95	$\mathbf{0.955}_{0.004}$	$\mathbf{0.010}_{0.001}$	$\mathbf{0.029}_{0.002}$	$\mathbf{0.857}_{0.012}$	$\mathbf{0.056}_{0.003}$	$\mathbf{0.185}_{0.006}$	$\mathbf{0.059}_{0.003}$	**15.664**

Table 16: Comparison f-divergences frontier summaries for the web text domain. The correlations from this table are reported in Table 9 of Section 7.4. The subscripts denote standard deviations over 5 random seeds. Boldfaced numbers indicate the smallest gap between the two distributions.

D.3 Comparison with Other Divergences and Optimal Transport

The full version of Tables 9 and 10 from Section 7.4 are given as Tables 16 and 17 respectively.

D.4 Use of Mauve for Hyperparameter Tuning

Figure 23 plots Mauve for nucleus and top-K sampling for various values of the hyperparameters p and K. This illustrates the utility of Mauve for hyperparameter tuning.

Appendix E. Human Evaluation: Protocol and Full Results

We describe the human evaluation protocol and results of Section 6.4 in detail. The outline for this section is:

- Appendix E.1: Details of the Bradley-Terry statistical model used to obtain ranking from pairwise preferences.
- Appendix E.2: Full results of the human evaluation.

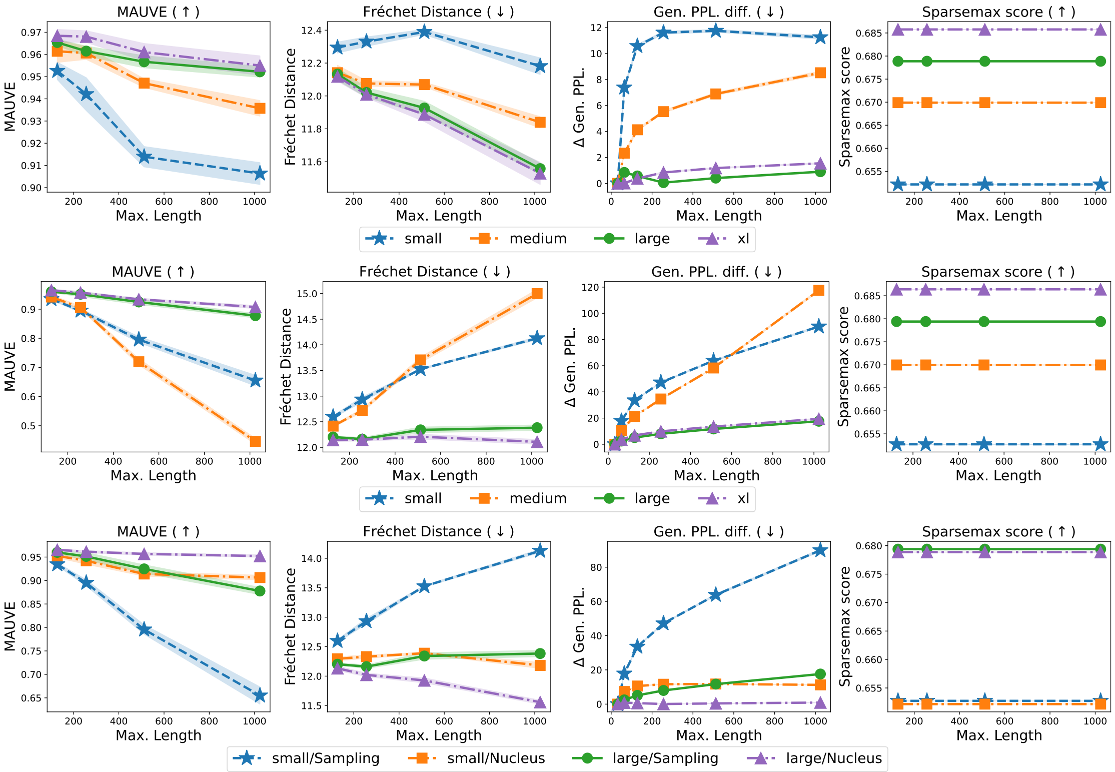


Figure 22: Generation quality versus maximum generation length as per various comparison measures for web text generation with GPT-2. We expect the quality of the generation to degrade as the maximum length of the text (both machine and human-written) increases. Mauve is the only comparison measure that correctly shows this behavior across all models and decoding algorithms. The shaded area denotes one standard deviation over generations from 5 random seeds.

GPT-2 Size	**Decoding**	**OT variants** (↓)			**Mauve variants** (↑)			**BT** (↑) **Human-like**
		Plug-in	Fréchet	Quantized	OT + Linear interpolation	OT + Barycenteric interpolation	(Default) KL + Linear interpolation	
small	Sampling	$763.281_{1.264}$	$199.591_{0.788}$	$0.158_{0.001}$	$0.814_{0.001}$	$0.937_{0.001}$	$0.655_{0.018}$	−27.518
	Nucleus, 0.9	$693.263_{4.610}$	$148.388_{1.236}$	$0.083_{0.005}$	$0.935_{0.007}$	$0.964_{0.001}$	$0.906_{0.005}$	−15.783
medium	Sampling	$791.758_{4.780}$	$224.970_{2.526}$	$0.208_{0.002}$	$0.725_{0.004}$	$0.914_{0.001}$	$0.446_{0.010}$	−30.769
	Nucleus, 0.9	$700.496_{3.961}$	$140.174_{0.813}$	$0.077_{0.004}$	$0.942_{0.005}$	$0.967_{0.001}$	$0.936_{0.004}$	−3.429
large	Sampling	$717.909_{5.618}$	$153.358_{1.325}$	$0.104_{0.004}$	$0.905_{0.006}$	$0.958_{0.002}$	$0.878_{0.008}$	−6.935
	Nucleus, 0.95	$\mathbf{681.883}_{4.367}$	$133.583_{0.762}$	$0.062_{0.001}$	$0.961_{0.002}$	$0.969_{0.001}$	$0.952_{0.002}$	12.553
xl	Sampling	$705.482_{4.617}$	$146.593_{1.136}$	$0.090_{0.003}$	$0.924_{0.004}$	$0.962_{0.001}$	$0.908_{0.005}$	8.966
	Nucleus, 0.95	$685.131_{3.258}$	$\mathbf{132.927}_{1.555}$	$\mathbf{0.061}_{0.003}$	$\mathbf{0.962}_{0.004}$	$\mathbf{0.970}_{0.001}$	$\mathbf{0.955}_{0.004}$	**15.664**

Table 17: Comparison of measures based on optimal transport for the web text domain. The correlations from this table are reported in Table 10 of Section 7.4. The subscripts denote standard deviations over 5 random seeds. Boldfaced numbers indicate the smallest gap between the two distributions.

- Appendix E.3: Additional details of the human evaluation protocol.

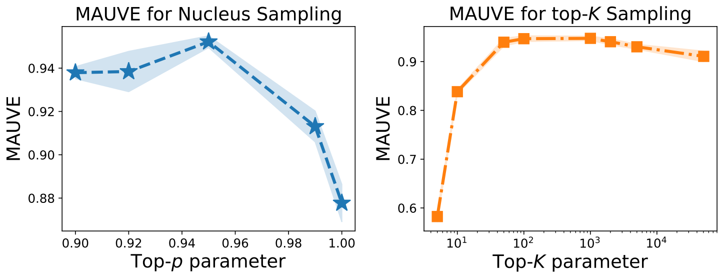
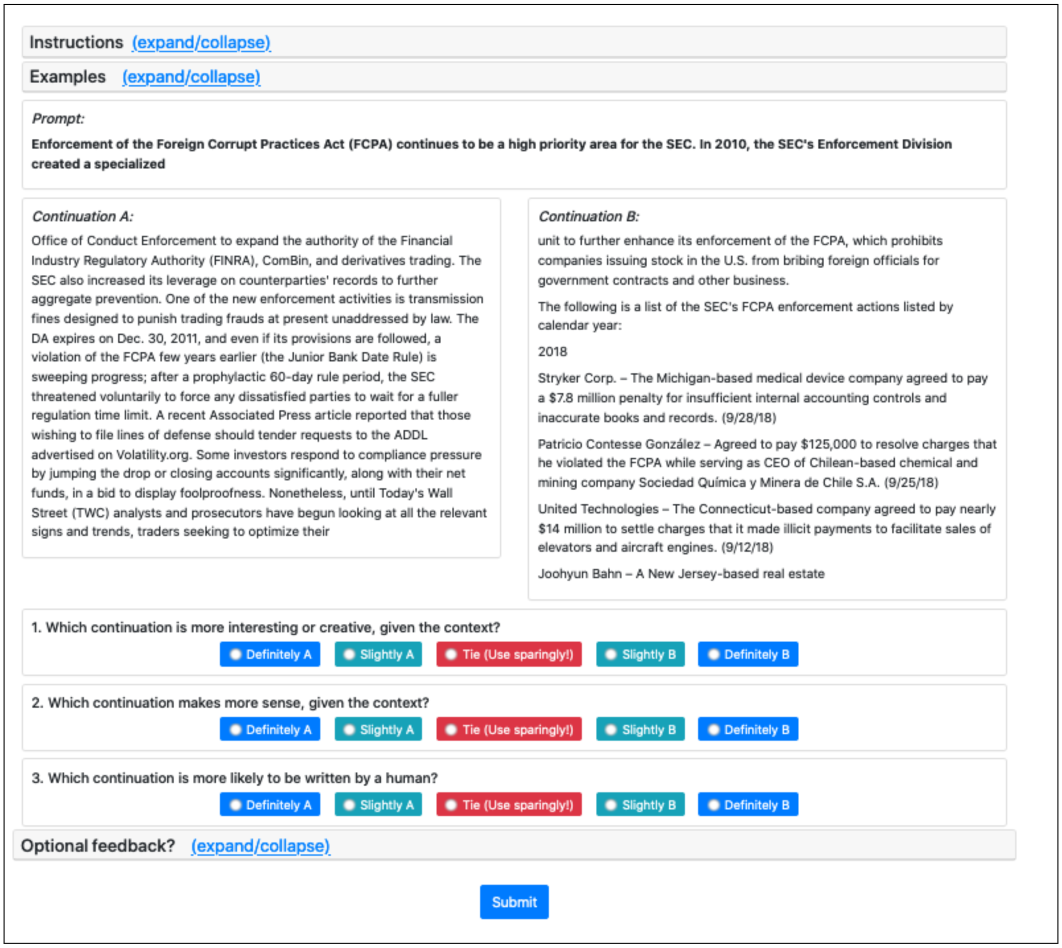


Figure 23: MAUVE for nucleus and top-K sampling for different values of p and K for GPT-2 large. MAUVE rates nucleus sampling with $p = 0.95$ and top-K sampling with $100 \leq K \leq 1000$ as the best choices. The shaded area denotes one standard deviation over generations from 5 random seeds.



Instructions (expand/collapse)

Examples (expand/collapse)

Prompt:

Enforcement of the Foreign Corrupt Practices Act (FCPA) continues to be a high priority area for the SEC. In 2010, the SEC's Enforcement Division created a specialized

Continuation A:

Office of Conduct Enforcement to expand the authority of the Financial Industry Regulatory Authority (FINRA), ComBin, and derivatives trading. The SEC also increased its leverage on counterparties' records to further aggregate prevention. One of the new enforcement activities is transmission fines designed to punish trading frauds at present unaddressed by law. The DA expires on Dec. 30, 2011, and even if its provisions are followed, a violation of the FCPA few years earlier (the Junior Bank Date Rule) is sweeping progress; after a prophylactic 60-day rule period, the SEC threatened voluntarily to force any dissatisfied parties to wait for a fuller regulation time limit. A recent Associated Press article reported that those wishing to file lines of defense should tender requests to the ADDL advertised on Volatility.org. Some investors respond to compliance pressure by jumping the drop or closing accounts significantly, along with their net funds, in a bid to display foolproofness. Nonetheless, until Today's Wall Street (TWC) analysts and prosecutors have begun looking at all the relevant signs and trends, traders seeking to optimize their

Continuation B:

unit to further enhance its enforcement of the FCPA, which prohibits companies issuing stock in the U.S. from bribing foreign officials for government contracts and other business.

The following is a list of the SEC's FCPA enforcement actions listed by calendar year:

2018

Stryker Corp. – The Michigan-based medical device company agreed to pay a $7.8 million penalty for insufficient internal accounting controls and inaccurate books and records. (9/28/18)

Patricio Contesse González – Agreed to pay $125,000 to resolve charges that he violated the FCPA while serving as CEO of Chilean-based chemical and mining company Sociedad Química y Minera de Chile S.A. (9/25/18)

United Technologies – The Connecticut-based company agreed to pay nearly $14 million to settle charges that it made illicit payments to facilitate sales of elevators and aircraft engines. (9/12/18)

Joohyun Bahn – A New Jersey-based real estate

1. Which continuation is more interesting or creative, given the context?

○ Definitely A ○ Slightly A ○ Tie (Use sparingly!) ○ Slightly B ○ Definitely B

2. Which continuation makes more sense, given the context?

○ Definitely A ○ Slightly A ○ Tie (Use sparingly!) ○ Slightly B ○ Definitely B

3. Which continuation is more likely to be written by a human?

○ Definitely A ○ Slightly A ○ Tie (Use sparingly!) ○ Slightly B ○ Definitely B

Optional feedback? (expand/collapse)

Submit

Figure 24: Mechanical Turk interface for human evaluation.

E.1 From Pairwise Preferences to Ranking: the Bradley-Terry Model

We compute the Bradley-Terry (BT) scores from the pairwise preferences obtained from the human evaluation along each of the three axes interesting, sensible, and more likely to be written by a human.

Bradley-Terry Model Review. Given n players with scores $w_1, \cdots, w_n$, the the Bradley-Terry model (Marden, 1995) models the outcome of a head-to-head comparison of any two players using a sigmoid[13]

$$\mathrm{Prob}(i \text{ beats } j) = \frac{1}{1 + e^{-(w_i - w_j)/100}} \,.$$

The model also assumes the outcome of each head-to-head comparison of any pair of players is independent of all other comparisons. Note that the model is invariant to additive shifts of the scores, i.e., the model probabilities induced by scores $w_1 + C, \cdots, w_n + C$ is same as the that induced by $w_1, \cdots, w_n$ for any constant C. For uniqueness, we normalize the scores so that their mean is 0.

Fitting the Model. The Bradley-Terry model can be fit to data using Zermelo's algorithm (Hunter, 2004). Suppose that we are given a dataset of head-to-head comparisons summarized by numbers N_{ij} denoting the number of times player i has defeated player j. Then, the negative log-likelihood $\ell(w_1, \cdots w_n)$ of the data under the The Bradley-Terry model can be written as

$$\ell(w_1, \cdots, w_n) = -\sum_{i=1}^n \sum_{j=1}^n N_{ij} \log(1 + e^{-(w_i - w_j)/100}) \,.$$

This is convex in the parameters $w_1, \cdots, w_n$ since the log-sum-exp function is convex. Zermelo's algorithm (Hunter, 2004) can be used to compute the maximum likelihood estimate. Denote $\widetilde{w}_i = w_i/100$. Starting from an initial estimate $\widetilde{w}_1^{(0)}, \cdots, \widetilde{w}_n^{(0)}$, each iteration of Zermelo's algorithm performs the update

$$u_i^{(t)} = \log\left(\sum_{j \ne i} N_{ij}\right) - \log\left(\sum_{j \ne i} \frac{N_{ij} + N_{ji}}{\exp(\widetilde{w}_i^{(t)}) + \exp(\widetilde{w}_j^{(t)})}\right)$$

followed by the mean normalization

$$\widetilde{w}_i^{(t+1)} = u_i^{(t)} - \frac{1}{n}\sum_{j=1}^n u_j^{(t)} \,.$$

Processing Raw Data. We collect the result of a head-to-head comparison using 5 options: Definitely A/B, Slightly A/B, or a Tie. We combine “Definitely A” and ”Slightly A“ into a single category denoting that A wins, while ties were assigned to either A or B uniformly at random.

[13] The scaling factor 100 is arbitrary and does not change the model or the obtained rankings.

GPT-2 Size	Decoding	BT/Human-like	BT/Interesting	BT/Sensible
Human		47.251	25.503	43.229
xl	Nucleus, $p = 0.95$	**15.664**	**23.046**	**31.888**
	Sampling	8.966	9.529	7.753
large	Nucleus, $p = 0.95$	12.553	6.785	8.781
	Sampling	−6.935	−1.532	−7.106
medium	Nucleus, $p = 0.9$	−3.429	−12.824	−7.293
	Sampling	−30.769	−34.323	−32.004
small	Nucleus, $p = 0.9$	−15.783	−0.697	−7.442
	Sampling	−27.518	−15.487	−37.805

Table 18: Fitted Bradley-Terry (BT) scores for each of the three axes rated by human annotators: "Human-like" denotes measures how likely the text is to be written by a human, while "Interesting" and "Sensible" quantify how interesting or sensible the text is. The Spearman rank correlations between each of these scores are: Human-like and Interesting: **0.917**, Human-like and Sensible: **0.917**, Interesting and Sensible: **0.967**.

E.2 Full Results of the Human Evaluation

BT Model for Human Eval. In our setting, each "player" is a source of text, i.e., one human, plus, eight model and decoding algorithm pairs (four model sizes GPT-2 small-/medium/large/xl coupled with pure sampling or nucleus sampling). We compute the BT score of each player as the maximum likelihood estimate of corresponding the parameters $w_1, \cdots, w_n$ based on head-to-head human evaluation data.

A higher BT score indicates a stronger preference from human annotators. The BT scores are reported in Table 18.

Interpreting BT scores. The BT scores reported in Table 18 give us predictions from the sigmoid model above. For example, consider the column "BT/Human-like". The best model-generated text, GPT-2 xl with nucleus sampling will lose to human text with probability 0.578. At the other end, GPT-2 small with nucleus sampling will lose to human text with probability 0.679. This shows that there is still much room for improvement in model-generated text.

Discussion. In general, the BT scores from human evaluations and Mauve both indicate that (a) nucleus sampling is better than pure sampling for the same model size, and, (b) larger model sizes are better for the same decoding algorithm. There is one exception to this rule, as per both the human evaluations and Mauve: GPT-2 small is better than GPT-2 medium for pure sampling.

E.3 Additional Details

We describe more details for the human evaluation. The terminology below is taken from (Shimorina and Belz, 2022).

Number of Outputs Evaluated. We compare 9 players: one player is "human", representing human-written text, whereas the other 8 are text generated by the model using the first 35 tokens of the corresponding human generation as a prompt. Each of the 8 non-human players come from a GPT-2 model of different sizes (small, medium, large, xl) and two decoding algorithms (pure sampling and nucleus sampling). We perform 90 comparisons between each pair of players, so each player is evaluated $90 \times 8 = 720$ times.

Prompt Filtering. We manually selected 1831 out of 5000 prompts which are well-formed English sentences from the web text test set.[14] For every head-to-head comparison, we sample 90 prompt without replacement and then sample the corresponding completions (for human-generated text, we use the test set of web text). We only consider a pair of players for human evaluation if the generation from each player is at least 200 BPE tokens long (and we truncate each generation at a maximum length of 256 BPE tokens).

Number of Evaluators. 214 unique evaluators participated in the evaluation. Of these, 11 evaluators supplied at least 50 annotations 95 evaluators supplied at least 10 annotations.

Evaluator Selection and Pay. We conduct our human evaluation on Amazon Mechanical Turk. Since the task only requires elementary reading and understanding skills in English, we open the evaluations to non-experts. Each crowd worker was paid 0.40 per annotation. The pay was estimated based on a $16/hour wage for the 85$^{\text{th}}$ percentile of response times from a pilot study (which was approx. 98 seconds per annotation). These evaluators are not previously known to the authors.

Training and Instructions. The evaluators were given instructions about the task and two detailed examples. No other training was provided due to the elementary nature of the task. The screenshots of these examples are given in Figure 25 while the instructions read:

> **Task Info**: We are studying how good AI models are at generating text on the internet. You are given a snippet of text from a random document on the internet, called the "prompt" or the "context", as well as two continuations, A and B. One or both of these is written by an AI. You must choose (a) which of two continuations is more interesting, (b) which makes more sense given the prompt, and, (c) which is more likely to have been written by a human, as per your assessment.
>
> **Guidelines**:
>
> - There are five choices for each question: Definitely A/B, Slightly A/B, or Tie. Please use the "Tie" option extremely sparingly! (No more than one in every ten pairs should be chosen as a tie along any of the three questions).
> - The questions can have different answers! Some text is very creative or interesting, but it doesn't quite fit the prompt or make sense.
> - Try to focus on quality over quantity. The text can be long but contain rambly gibberish.
> - Don't worry if the text ends abruptly or has other artifacts of the website downloading process (text like 'Advertisement' for instance).
> - Please do your best, some of these are pretty challenging!
> - Answering each question should take around 1.5 minutes on average, as per our estimation. We have calibrated the pay to be $16 per hour with this speed.

[14] The web text dataset is scraped from the internet and is *not* curated. This dataset contains poor prompts such as headers of webpages or error messages, such as: “Having trouble viewing the video? Try disabling any ad-blocking extensions currently running on your browser” or “Front Page Torrents Favorites My Home My Galleries Toplists Bounties News Forums Wiki”. We exclude such prompts as they are unsuitable for human evaluation.

Quality Control. All annotations made in under 25 seconds were excluded for quality control (the mean response time per annotation was 47 seconds).

Quality Criteria. We use three quality criteria. The questions asked to the evaluators are (verbatim):

1. Interestingness: “Which continuation is more interesting or creative, given the context?"
2. Sensible: “Which continuation makes more sense, given the context?”
3. Human-like: “Which continuation is more likely to be written by a human?”

Note that we do explicitly name the criteria in the evaluation form, although those names could be inferred from the definitions. We use these names only in the paper.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

Morteza Alamgir, Gábor Lugosi, and Ulrike von Luxburg. Density-preserving quantization with application to graph downsampling. In *COLT*, 2014.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. Falcon-40B: an open large language model with state-of-the-art performance. 2023.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, 2005.

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. Mirostat: A Neural Text Decoding Algorithm that Directly Controls Perplexity. In *Proc. of ICLR*, 2021.

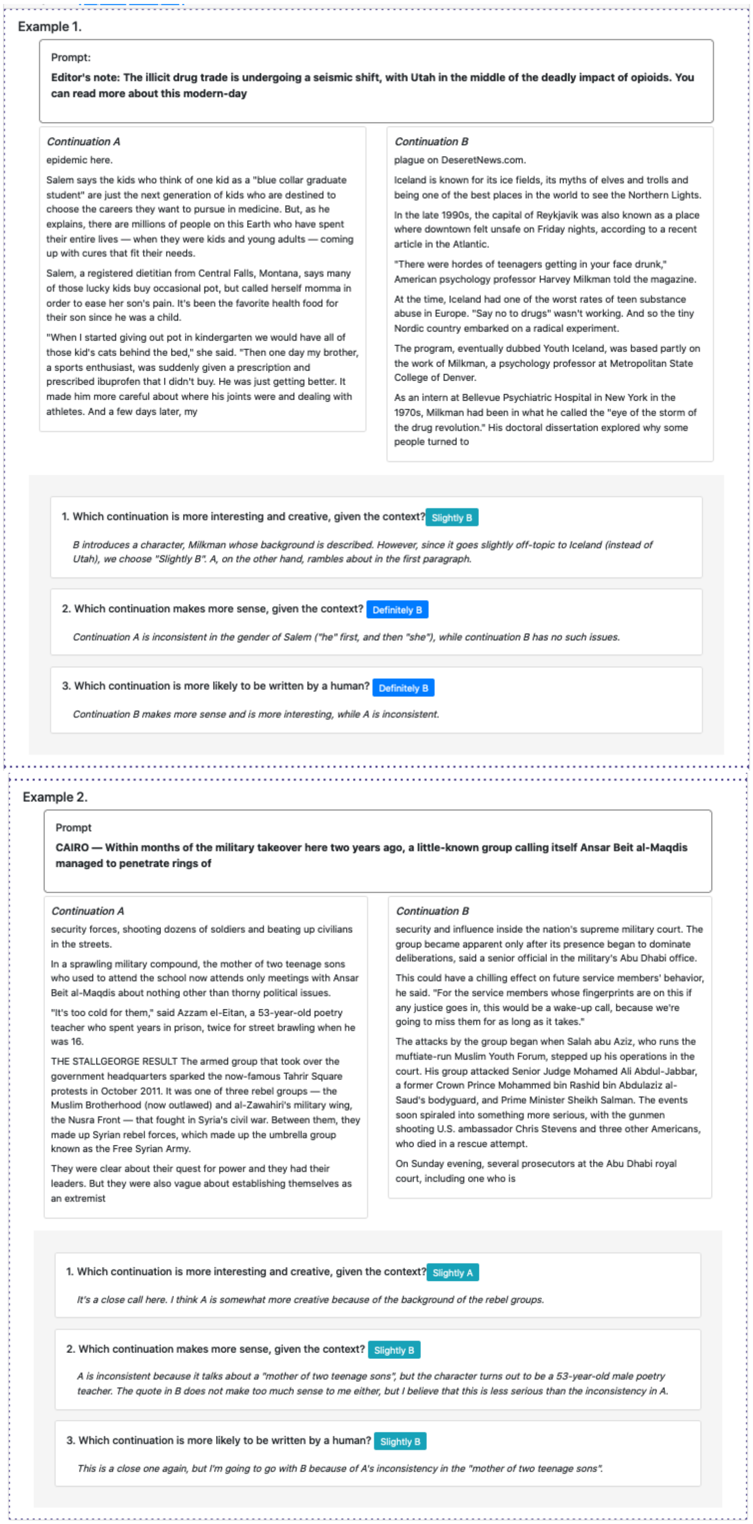
Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003.

Daniel Berend and Aryeh Kontorovich. The missing mass problem. *Statistics & Probability Letters*, 82(6), 2012.

Gérard Biau, Maxime Sangnier, and Ugo Tanielian. Some theoretical insights into Wasserstein GANs. *Journal of Machine Learning Research*, 2021.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *Proc. of ICLR*, 2018.

Dietrich Braess and Tomas Sauer. Bernstein polynomials and learning theory. *Journal of Approximation Theory*, 128(2), 2004.



Example 1.

Prompt:

Editor's note: The illicit drug trade is undergoing a seismic shift, with Utah in the middle of the deadly impact of opioids. You can read more about this modern-day

Continuation A

epidemic here.

Salem says the kids who think of one kid as a "blue collar graduate student" are just the next generation of kids who are destined to choose the careers they want to pursue in medicine. But, as he explains, there are millions of people on this Earth who have spent their entire lives — when they were kids and young adults — coming up with cures that fit their needs.

Salem, a registered dietitian from Central Falls, Montana, says many of those lucky kids buy occasional pot, but called herself momma in order to ease her son's pain. It's been the favorite health food for their son since he was a child.

"When I started giving out pot in kindergarten we would have all of those kid's cats behind the bed," she said. "Then one day my brother, a sports enthusiast, was suddenly given a prescription and prescribed ibuprofen that I didn't buy. He was just getting better. It made him more careful about where his joints were and dealing with athletes. And a few days later, my

Continuation B

plague on DeseretNews.com.

Iceland is known for its ice fields, its myths of elves and trolls and being one of the best places in the world to see the Northern Lights.

In the late 1990s, the capital of Reykjavik was also known as a place where downtown felt unsafe on Friday nights, according to a recent article in the Atlantic.

"There were hordes of teenagers getting in your face drunk," American psychology professor Harvey Milkman told the magazine.

At the time, Iceland had one of the worst rates of teen substance abuse in Europe. "Say no to drugs" wasn't working. And so the tiny Nordic country embarked on a radical experiment.

The program, eventually dubbed Youth Iceland, was based partly on the work of Milkman, a psychology professor at Metropolitan State College of Denver.

As an intern at Bellevue Psychiatric Hospital in New York in the 1970s, Milkman had been in what he called the "eye of the storm of the drug revolution." His doctoral dissertation explored why some people turned to

1. Which continuation is more interesting and creative, given the context? Slightly B

B introduces a character, Milkman whose background is described. However, since it goes slightly off-topic to Iceland (instead of Utah), we choose "Slightly B". A, on the other hand, rambles about in the first paragraph.

2. Which continuation makes more sense, given the context? Definitely B

Continuation A is inconsistent in the gender of Salem ("he" first, and then "she"), while continuation B has no such issues.

3. Which continuation is more likely to be written by a human? Definitely B

Continuation B makes more sense and is more interesting, while A is inconsistent.

Example 2.

Prompt

CAIRO — Within months of the military takeover here two years ago, a little-known group calling itself Ansar Beit al-Maqdis managed to penetrate rings of

Continuation A

security forces, shooting dozens of soldiers and beating up civilians in the streets.

In a sprawling military compound, the mother of two teenage sons who used to attend the school now attends only meetings with Ansar Beit al-Maqdis about nothing other than thorny political issues.

"It's too cold for them," said Azzam el-Eitan, a 53-year-old poetry teacher who spent years in prison, twice for street brawling when he was 16.

THE STALLGEORGE RESULT The armed group that took over the government headquarters sparked the now-famous Tahrir Square protests in October 2011. It was one of three rebel groups — the Muslim Brotherhood (now outlawed) and al-Zawahiri's military wing, the Nusra Front — that fought in Syria's civil war. Between them, they made up Syrian rebel forces, which made up the umbrella group known as the Free Syrian Army.

They were clear about their quest for power and they had their leaders. But they were also vague about establishing themselves as an extremist

Continuation B

security and influence inside the nation's supreme military court. The group became apparent only after its presence began to dominate deliberations, said a senior official in the military's Abu Dhabi office.

This could have a chilling effect on future service members' behavior, he said. "For the service members whose fingerprints are on this if any justice goes in, this would be a wake-up call, because we're going to miss them for as long as it takes."

The attacks by the group began when Salah abu Aziz, who runs the muftiate-run Muslim Youth Forum, stepped up his operations in the court. His group attacked Senior Judge Mohamed Ali Abdul-Jabbar, a former Crown Prince Mohammed bin Rashid bin Abdulaziz al-Saud's bodyguard, and Prime Minister Sheikh Salman. The events soon spiraled into something more serious, with the gunmen shooting U.S. ambassador Chris Stevens and three other Americans, who died in a rescue attempt.

On Sunday evening, several prosecutors at the Abu Dhabi royal court, including one who is

1. Which continuation is more interesting and creative, given the context? Slightly A

It's a close call here. I think A is somewhat more creative because of the background of the rebel groups.

2. Which continuation makes more sense, given the context? Slightly B

A is inconsistent because it talks about a "mother of two teenage sons", but the character turns out to be a 53-year-old male poetry teacher. The quote in B does not make too much sense to me either, but I believe that this is less serious than the inconsistency in A.

3. Which continuation is more likely to be written by a human? Slightly B

This is a close one again, but I'm going to go with B because of A's inconsistency in the "mother of two teenage sons".

Figure 25: Annotated examples shown to the evaluators.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Proc. of NeurIPS*, 2020.

Yuheng Bu, Shaofeng Zou, Yingbin Liang, and Venugopal V. Veeravalli. Estimation of KL divergence: Optimal minimax rate. *IEEE Transactions on Information Theory*, 64(4), 2018.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. Language GANs Falling Short. In *Proc. of ICLR*, 2020.

Haixiao Cai, Sanjeev R. Kulkarni, and Sergio Verdú. Universal divergence estimation for finite-alphabet sources. *IEEE Trans. Inf. Theory*, 52(8):3456–3475, 2006.

T. Tony Cai, X. Jessie Jeng, and Jiashun Jin. Optimal detection of heterogeneous and heteroscedastic mixtures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5), 2011.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. Evaluation of Text Generation: A Survey. *arXiv Preprint*, 2020.

David M Chan, Yiming Ni, Austin Myers, Sudheendra Vijayanarasimhan, David A Ross, and John Canny. Distribution aware metrics for conditional natural language generation. *arXiv preprint arXiv:2209.07518*, 2022.

Fasil Cheema and Ruth Urner. Precision Recall Cover: A Method For Assessing Generative Models. In *AISTATS*, volume 206, pages 6571–6594, 2023.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4), 1999.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. Sentence Mover's Similarity: Automatic Evaluation for Multi-Sentence Texts. In *Proc. of ACL*, 2019.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text. In *Proc. of ACL*, 2021.

Stéphan Clémençon and Nicolas Vayatis. Nonparametric estimation of the precision-recall curve. In *Proc. of ICML*, pages 185–192, 2009.

Stéphan Clémençon and Nicolas Vayatis. Overlaying classifiers: a practical approach to optimal scoring. *Constructive Approximation*, 32:619–648, 2010.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011.

Corinna Cortes and Mehryar Mohri. Confidence intervals for the area under the ROC curve. In *Proc. of NeurIPS*, volume 17, 2005.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Proc. of NIPS*, 2013.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL*, pages 4171–4186, 2019.

Luc Devroye, Lászl Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

Josip Djolonga, Mario Lucic, Marco Cuturi, Olivier Bachem, Olivier Bousquet, and Sylvain Gelly. Precision-Recall Curves Using Information Divergence Frontiers. In *Proc. of AISTATS*, pages 2550–2559, 2020.

Bryan Eikema and Wilker Aziz. Is MAP Decoding All You Need? The Inadequacy of the Mode in Neural Machine Translation. In *Proc. of CoLING*, 2020.

Moein Falahatgar, Mesrob I Ohannessian, Alon Orlitsky, and Venkatadheeraj Pichapati. The power of absolute discounting: All-dimensional distribution estimation. In *Proc. of NIPS*, 2017.

Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical Neural Story Generation. In *Proc. of ACL*, pages 889–898, 2018.

Matthew Finlayson, John Hewitt, Alexander Koller, Swabha Swayamdipta, and Ashish Sabharwal. Closing the curious case of neural text degeneration, 2023.

Peter Flach. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press, 2012.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In *EMNLP*, pages 3356–3369, 2020.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text. *J. Artif. Intell. Res.*, 77:103–166, 2023.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4), 1953.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. In *Proc. of NeurIPS*, 2014.

Google. Bard: A conversational AI tool by Google. https://bard.google.com, 2023.

Jian Guan and Minlie Huang. UNION: An Unreferenced Metric for Evaluating Open-ended Story Generation. In *Proc. of EMNLP*, pages 9157–9166, 2020.

Adityanand Guntuboyina, Sujayam Saha, and Geoffrey Schiebinger. Sharp Inequalities for f-Divergences. *IEEE Trans. Inf. Theory*, 60(1):104–121, 2014.

L. Györfi and T. Nemetz. f-dissimilarity: A generalization of the affinity of several distributions. *Annals of the Institute of Statistical Mathematics*, 30, 1978.

Perttu Hämäläinen and Arno Solin. Deep Residual Mixture Models. *arXiv preprint*, 2020.

Yanjun Han, Jiantao Jiao, and Tsachy Weissman. Minimax Estimation of Divergences Between Discrete Distributions. *IEEE J. Sel. Areas Inf. Theory*, 1(3):814–823, 2020.

Tatsunori Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation. In *Proc. of NAACL*, pages 1689–1701, 2019.

Anant Hegde, Deniz Erdogmus, Tue Lehn-Schioler, Yadunandana N Rao, and Jose C Principe. Vector-quantization by density matching in the minimum Kullback-Leibler divergence sense. In *IJCNN*, 2004.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Proc. of NeurIPS*, 30, 2017.

John Hewitt, Christopher D Manning, and Percy Liang. Truncation Sampling as Language Model Desmoothing. In *Proc. of EMNLP Findings*, 2022.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. In *Proc. of ICLR*, 2020.

Jinyi Hu, Xiaoyuan Yi, Wenhao Li, Maosong Sun, and Xing Xie. Fuse It More Deeply! A Variational Transformer with Layer-Wise Latent Variable Inference for Text Generation. In *Proc. of NAACL*, 2022.

Marc Van Hulle. Faithful representations with topographic maps. *Neural Networks*, 12(6), 1999.

David R Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1):384–406, 2004.

Yuri Ingster and I.A. Suslina. *Nonparametric Goodness-of-Fit Testing Under Gaussian Models*. Springer, 2003.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In *Proc. of ACL*, pages 1808–1822, July 2020.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *ArXiv Preprint*, 2023.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.

Dan Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, Pearson Education International, 2009.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry Wasserman, et al. Nonparametric von Mises Estimators for Entropies, Divergences and Mutual Informations. *Advances in Neural Information Processing Systems*, 28, 2015.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. The Perils of Using Mechanical Turk to Evaluate Open-Ended Text Generation. In *Proc. of EMNLP*, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proc. of CVPR*, pages 4401–4410, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training Generative Adversarial Networks with Limited Data. *Proc. of NeurIPS*, 33: 12104–12114, 2020a.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *Proc. of CVPR*, pages 8107–8116, 2020b.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-Free Generative Adversarial Networks. In *Proc. of NeurIPS*, pages 852–863, 2021.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet Audio Distance: A Reference-Free Metric for Evaluating Music Enhancement Algorithms. In *Interspeech*, pages 2350–2354, 2019.

Pum Jun Kim, Yoojin Jang, Jisu Kim, and Jaejun Yoo. TopP&R: Robust Support Estimation Approach for Evaluating Fidelity and Diversity in Generative Models. In *Proc. of NeurIPS*, 2023.

George Kour, Samuel Ackerman, Eitan Farchi, Orna Raz, Boaz Carmeli, and Ateret Anaby-Tavor. Measuring the Measuring Tools: An Automatic Evaluation of Semantic Metrics for Text Corpora. In *Proc. of EMNLP*, 2022.

Raphail E. Krichevsky and Victor K. Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2), 1981.

Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. Harnessing large-language models to generate private synthetic text. *arXiv Preprint*, 2023.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From Word Embeddings to Document Distances. In *Proc. of ICML*, pages 957–966. PMLR, 2015.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved Precision and Recall Metric for Assessing Generative Models. In *Proc. of NeurIPS*, 2019.

Tian Lan, Yixuan Su, Shuhang Liu, Heyan Huang, and Xian-Ling Mao. Momentum Decoding: Open-ended Text Generation As Graph Exploration. *arXiv Preprint*, 2022.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive Decoding: Open-ended Text Generation as Optimization. In *Proc. of ACL*, pages 12286–12312, 2023.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, et al. Holistic evaluation of language models, 2023.

Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, 2004.

Lang Liu, Krishna Pillutla, Sean Welleck, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. Divergence Frontiers for Generative Models: Sample Complexity, Quantization Effects, and Frontier Integrals. In *Proc. of NeurIPS*, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv Preprint*, 2019.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text Classification using String Kernels. *Journal of machine learning research*, 2(Feb): 419–444, 2002.

David Lopez-Paz and Maxime Oquab. Revisiting Classifier Two-Sample Tests. In *Proc. of ICLR*, 2017.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2001. ISBN 978-0-262-13360-9.

John I. Marden. *Analyzing and modeling rank data*, volume 64 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 1995. ISBN 0-412-99521-2.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. Sparse Text Generation. In *Proc. EMNLP*, pages 4252–4273, 2020.

Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. How Decoding Strategies Affect the Verifiability of Generated Text. In *Proc. of EMNLP*, pages 223–235, 2020.

Justus Mattern, Zhijing Jin, Benjamin Weggenmann, Bernhard Schoelkopf, and Mrinmaya Sachan. Differentially Private Language Models for Secure Data Sharing. In *Proc. of EMNLP*, 2022.

David Mcallester and Luis Ortiz. Concentration inequalities for the missing mass and for histogram rule error. *Journal of Machine Learning Research*, 4, 2003.

Peter Meinicke and Helge Ritter. Quantizing density estimators. In *Proc. of NeurIPS*, 2002.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally Typical Sampling. *Transactions of the Association for Computational Linguistics*, 2022.

James Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London (A)*, 209 (441-458):415–446, 1909.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Springer Science & Business Media, 2012.

Kevin R. Moon and Alfred O. Hero III. Ensemble estimation of multivariate f-divergence. In *Proc. of ISIT*, pages 356–360. IEEE, 2014.

XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. Estimating Divergence Functionals and the Likelihood Ratio by Convex Risk Minimization. *IEEE Trans. Inf. Theory*, 56(11):5847–5861, 2010.

Frank Nielsen and Rajendra Bhatia. *Matrix Information Geometry*. Springer, 2013.

Morteza Noshad, Kevin R. Moon, Salimeh Yasaei Sekeh, and Alfred O. Hero III. Direct estimation of information divergence using nearest neighbor ratios. In *ISIT*, pages 903–907. IEEE, 2017.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why We Need New Evaluation Metrics for NLG. In *Proc. of EMNLP*, 2017.

OpenAI. GPT-4 Technical Report, 2023.

Juri Opitz and Anette Frank. Towards a Decomposable Metric for Explainable Evaluation of Text Generation from AMR. In *Proc. of EACL*, pages 1504–1518, 2021.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *CoRR*, 2023.

Alon Orlitsky and Ananda Theertha Suresh. Competitive distribution estimation: Why is Good-Turing good. In *NeurIPS*, 2015.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*, pages 311–318, 2002.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global Vectors for Word Representation. In *Proc. of EMNLP*, pages 1532–1543, 2014.

Margaret Sullivan Pepe. Receiver Operating Characteristic Methodology. *Journal of the American Statistical Association*, 95(449):308–311, 2000. ISSN 01621459.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the Gap Between Neural Text and Human Text with Divergence Frontiers. In *Proc. of NeurIPS*, 2021.

Tiago Pimentel, Clara Meister, and Ryan Cotterell. On the Usefulness of Embeddings, Clusters and Strings for Text Generation Evaluation. In *Proc. of ICLR*, 2023.

Barnabás Póczos, Liang Xiong, and Jeff G. Schneider. Nonparametric Divergence Estimation with Applications to Machine Learning on Distributions. In *Proc. of UAI*, pages 599–608, 2011.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9, 2019.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv Preprint*, 2022.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. PlotMachines: Outline-Conditioned Generation with Dynamic Plot State Tracking. In *Proc. of EMNLP*, pages 4274–4295, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proc. of CVPR*, pages 10684–10695, 2022.

Juho Rousu, John Shawe-Taylor, and Tommi Jaakkola. Efficient Computation of Gapped Substring Kernels on Large Alphabets. *Journal of Machine Learning Research*, 6(9), 2005.

Paul K. Rubenstein, Olivier Bousquet, Josip Djolonga, Carlos Riquelme, and Ilya O. Tolstikhin. Practical and Consistent Estimation of f-Divergences. In *Proc. of NeurIPS*, pages 4072–4082, 2019.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. In *Proc. of ICLR*, 2019.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *Proc. of NeurIPS*, 2022.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. A Survey of Evaluation Metrics Used for NLG Systems. *ACM Comput. Surv.*, 55(2):26:1–26:39, 2023.

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Proc. of NeurIPS*, 2018.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved Techniques for Training GANs. In *Proc. of NeurIPS*, pages 2226–2234, 2016.

Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets. In *SIGGRAPH*, pages 49:1–49:10, 2022.

Nicolas Schreuder, Victor-Emmanuel Brunel, and Arnak Dalalyan. Statistical guarantees for generative models without domination. In *Algorithmic Learning Theory*, pages 1051–1071. PMLR, 2021.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. BLEURT: Learning Robust Metrics for Text Generation. In *Proc. of ACL*, pages 7881–7892, 2020.

Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. On Accurate Evaluation of GANs for Language Generation, 2018. arXiv Preprint.

Serge Sharoff. Know thy Corpus! Robust Methods for Digital Curation of Web corpora. In *Proc. of LREC*, pages 2453–2460. European Language Resources Association, 2020.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. RUSE: Regressor Using Sentence Embeddingsfor Automatic Machine Translation Evaluation. In *Proc. of Conference on Machine Translation*, pages 751–758, 2018.

Anastasia Shimorina and Anya Belz. The human evaluation datasheet: A template for recording details of human evaluation experiments in NLP. In *Proc. of Workshop on Human Evaluation of NLP Systems*, pages 54–75, 2022.

Jorge Silva and Shrikanth Narayanan. Universal consistency of data-driven partitions for divergence estimation. In *Proc. of ISIT*, 2007.

Jorge Silva and Shrikanth S Narayanan. Information divergence estimation based on data-dependent partitions. *Journal of Statistical Planning and Inference*, 140(11), 2010.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proc. of ICLR*, 2021.

Sreejith Sreekumar and Ziv Goldfeld. Neural Estimation of Statistical Divergences. *Journal of Machine Learning Research*, 23(126):1–75, 2022.

George Stein, Jesse C. Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Leigh Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L. Caterini, J. Eric T. Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Proc. of NeurIPS 2023*, 2023.

Yixuan Su and Jialu Xu. An Empirical Study On Contrastive Search And Contrastive Decoding For Open-ended Text Generation. *arXiv Preprint*, 2022.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A Contrastive Framework for Neural Text Generation. In *Proc. of NeurIPS*, 2022.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density Ratio Estimation in Machine Learning.* Cambridge University Press, 2012.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation. *ArXiv Preprint*, 2021.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. RUBER: An Unsupervised Method for Automatic Evaluation of Open-Domain Dialog Systems. In *Proc. of AAAI*, 2018.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *ArXiv Preprint*, 2023.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards Accurate Generative Models of Video: A New Metric & Challenges. *arXiv Preprint*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Proc. of NeurIPS*, pages 5998–6008, 2017.

Sergio Verdú. Empirical Estimation of Information Measures: A Literature Guide. *Entropy*, 21(8):720, 2019.

Alexandre Verine, Benjamin Negrevergne, Muni Sreenivas Pydi, and Yann Chevaleyre. Precision-Recall Divergence Optimization for Generative Modeling with GANs and Normalizing Flows. In *Proc. of NeurIPS*, 2023.

Cédric Villani. *Topics in Optimal Transportation.* American Mathematical Society, 2003.

Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. Divergence estimation of continuous distributions based on data-dependent partitions. *IEEE Transactions on Information Theory*, 51(9), 2005.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to Model the Tail. In *Proc. of NeurIPS*, 2017.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. Consistency of a Recurrent Language Model With Respect to Incomplete Decoding. In *Proc. of EMNLP*, pages 5553–5568, 2020a.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural Text Generation With Unlikelihood Training. In *Proc. of ICLR*, 2020b.

BigScience Workshop. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *ArXiv Preprint*, 2022.

Zonghan Yang, Xiaoyuan Yi, Peng Li, Yang Liu, and Xing Xie. Unified Detoxifying and Debiasing in Language Generation via Inference-time Adaptive Optimization. In *Proc. of ICLR*, 2023.

Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling Autoregressive Multi-Modal Models: Pretraining and Instruction Tuning. *ArXiv Preprint*, 2023.

Xiang Yue, Huseyin A. Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. Synthetic Text Generation with Differential Privacy: A Simple and Practical Recipe. In *Proc. of ACL*, pages 1321–1342, 2023.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending Against Neural Fake News. In *Proc. of NeurIPS*, 2019.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, pages 586–595, 2018.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *Proc. of ICLR*, 2020.

Zhiyi Zhang and Michael Grabchak. Nonparametric estimation of Küllback-Leibler divergence. *Neural Comput.*, 26(11):2570–2593, 2014.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. In *Proc. of EMNLP*, 2019.