

f -Divergence Minimization for Sequence-Level Knowledge Distillation

Yuqiao Wen^{1,*}, Zichao Li^{2,*}, Wenyu Du³, Lili Mou^{1,4}

¹Dept. Computing Science & Alberta Machine Intelligence Institute (Amii), University of Alberta

²Mila, McGill University ³The University of Hong Kong

⁴Canada CIFAR AI Chair, Amii *Equal contribution

yq.when@gmail.com, zichao.li@mila.quebec

wenyudu@yahoo.com, doublepower.mou@gmail.com

Abstract

Knowledge distillation (KD) is the process of transferring knowledge from a large model to a small one. It has gained increasing attention in the natural language processing community, driven by the demands of compressing ever-growing language models. In this work, we propose an f -DISTILL framework, which formulates sequence-level knowledge distillation as minimizing a generalized f -divergence function. We propose four distilling variants under our framework and show that existing SeqKD and ENGINE approaches are approximations of our f -DISTILL methods. We further derive step-wise decomposition for our f -DISTILL, reducing intractable sequence-level divergence to word-level losses that can be computed in a tractable manner. Experiments across four datasets show that our methods outperform existing KD approaches, and that our symmetric distilling losses can better force the student to learn from the teacher distribution.¹

1 Introduction

Increasingly large language models have continued to achieve state-of-the-art performance across various natural language generation tasks, such as data-to-text generation (Lebret et al., 2016; Li and Liang, 2021), summarization (Paulus et al., 2018; Zhang et al., 2020a), and dialogue generation (Li et al., 2016b; Zhang et al., 2020b). However, super-large language models are inaccessible to most users and researchers due to their prohibitively large model size, emphasizing the importance of high-performing, parameter-efficient small neural models.

A widely used approach to training small models is *knowledge distillation* (KD, Hinton et al., 2015), where the small model (known as the *student*) learns the knowledge from a much larger model (known as the *teacher*). KD has shown great

success in helping smaller models achieve competitive performance across a wide range of applications (Sun et al., 2019; Jiao et al., 2020; Shleifer and Rush, 2020).

Existing KD approaches can be categorized into two main branches: representation matching and distribution matching. The former aims to imitate the teacher’s real-valued intermediate-layer representations, say, with mean squared error (Sun et al., 2019; Jiao et al., 2020). Our work focuses on the latter, distribution matching, where the student model learns the teacher’s predictive distribution. Hinton et al. (2015) minimize the cross-entropy loss against the teacher-predicted soft labels, which is equivalent to minimizing the Kullback–Leibler (KL) divergence between the teacher and student. Kim and Rush (2016) propose SeqKD, arguing that KL divergence should be minimized at the sequence level for language models. However, such an approach tends to learn an overly smooth student distribution to cover the entire support of the teacher distribution due to the asymmetric nature of the KL divergence. This is often known as the *mode-averaging* problem (Figure 1a).

Tu et al. (2020) propose ENGINE, a non-autoregressive translation model that minimizes the energy function defined by the teacher’s output distribution. It can be shown that their objective is related to minimizing the reverse KL between the teacher and student (see Section 2.2). This, on the other hand, results in the *mode-collapsing* problem, where the student model is overly concentrated on certain high-probability regions of the teacher distribution (Figure 1b).

In this paper, we address knowledge distillation for text generation tasks, and propose f -DISTILL, a unified framework that formulates sequence-level knowledge distillation as minimizing f -divergence functions. Existing SeqKD (Kim and Rush, 2016) and ENGINE (Tu et al., 2020) methods are approximations of KL and reverse KL distillations

¹Our code is available at <https://github.com/MANGA-UOFA/fdistill>

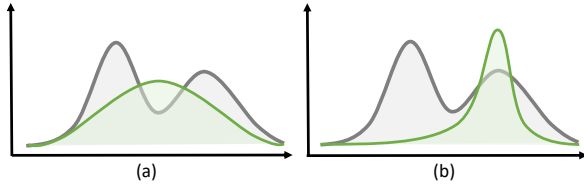


Figure 1: Typical phenomena of (a) KL distillation and (b) reverse KL distillation. Gray curve: teacher distribution. Green curve: student distribution.

under the f -DISTILL framework. Further, our formulation naturally leads to Jensen–Shannon (JS) divergence and total variation distance (TVD) distillations, where the divergence measures are symmetric in teacher and student distributions. This forces the student to learn the teacher’s distribution better, alleviating mode averaging and collapsing problems.

We further develop efficient algorithms for our f -DISTILL approach. First, we show that sequence-level f -divergence can be decomposed step by step either exactly or as an upper bound. Second, we propose to sample from the teacher model in an offline manner, mitigating the additional training cost of symmetric divergence measures (namely, JS and TVD).

We evaluated our approach on four datasets: DART for data-to-text generation (Nan et al., 2021), XSum for summarization (Narayan et al., 2018), WMT16 EN-RO for machine translation (Bojar et al., 2016), and Commonsense Dialogue (Zhou et al., 2021). Experiments show that our proposed f -DISTILL variants consistently outperform existing distribution-matching KD methods, allowing f -DISTILL to achieve an add-on performance improvement when combined with representation-matching KD methods. Further, results show that our symmetric distilling losses outperform asymmetric ones, confirming that extreme mode averaging or collapsing is not ideal.

To sum up, our contributions are three-fold:

1. We propose f -DISTILL, a novel distilling framework that generalizes KL distillation and balances mode averaging and collapsing;
2. We derive step-wise decomposition and propose an offline sampling method to efficiently compute sequence-level f -divergences; and
3. We provide detailed experimental analysis across four text generation datasets to show the effectiveness of our approach.

2 Approach

In this section, we first review classic knowledge distilling (KD) algorithms and analyze their drawbacks. Then, we propose f -DISTILL, a generalized distilling framework for sequence-level distillation.

2.1 Classic KD and Its Drawbacks

In classic KD, the KL divergence is often used to train the student model to match the teacher’s distribution (Hinton et al., 2015). For autoregressive text generation, this is decomposed into a step-wise KL divergence:

$$J_{\text{KD}} = - \sum_{t=1}^{|\mathbf{y}|} \sum_{Y_t \in V} p(Y_t | \mathbf{y}_{<t}) \log q_{\theta}(Y_t | \mathbf{y}_{<t}) \quad (1)$$

where $\mathbf{y} = y_1 \cdots y_T$ is the ground-truth sequence and V is the vocabulary. p and q_{θ} are the predicted distributions of the teacher and student, respectively; they can be additionally conditioned on an input sequence \mathbf{x} , which is omitted here for simplicity. In Eqn. (1), we present the loss by a cross-entropy term, which only differs from the KL divergence $D_{\text{KL}}(p \| q_{\theta})$ by a constant.

Kim and Rush (2016) propose SeqKD and minimize cross-entropy loss at the sequence level as

$$J_{\text{SeqKD}} = \mathbb{E}_{\mathbf{Y} \sim p}[-\log q_{\theta}(\mathbf{Y})] \quad (2)$$

In practice, the expectation over the sentence space is intractable, so they approximate it with a hard sequence \mathbf{y} generated by beam search on the teacher model. Their loss is

$$\hat{J}_{\text{SeqKD}} = - \sum_{t=1}^{|\mathbf{y}|} \log q_{\theta}(y_t | \mathbf{y}_{<t}) \quad (3)$$

However, KL-based losses may cause the student model to learn an overly smooth function. This can be seen in Eqn. (3), where the loss term $-\log q_{\theta}(y_t | \mathbf{y}_{<t})$ goes to infinity when the student assigns a low probability to a teacher-generated token. As a result, minimizing KL forces the student model to spread its probability mass widely over the vocabulary. When the student has a limited model capacity, this further leads to the mode-averaging problem, where the learned distribution may not capture any mode of the teacher distribution, as shown in Figure 1a.

Divergence	$f(t)$
Kullback–Leibler (KL)	$t \log t$
Reverse KL (RKL)	$-\log t$
Jensen–Shannon (JS)	$-(t+1) \log(\frac{t+1}{2}) + t \log t$
Total variation distance (TVD)	$\frac{1}{2} t-1 $

Table 1: Common divergence functions and their corresponding choices of f .

2.2 Our Proposed f -DISTILL Framework

To this end, we propose a generalized f -DISTILL framework, a family of distilling methods based on f -divergence functions (Ali and Silvey, 1966; Sason and Verdú, 2016).

Formally, the f -divergence of two distributions is defined as

$$D_f(p(t)||q(t)) = \sum_t q(t) f\left(\frac{p(t)}{q(t)}\right) \quad (4)$$

where $f : (0, \infty) \rightarrow \mathbb{R}$ is a convex function such that $f(1) = 0$. Table 1 summarizes common divergence functions.

In the rest of this subsection, we will first present Kullback–Leibler (KL) and reverse KL (RKL) distilling methods, which are closely related to previous work (Kim and Rush, 2016; Tu et al., 2020). Then, we will propose Jensen–Shannon (JS) and total variation distance (TVD) distillations; they are based on symmetric f -divergence functions, and are able to force the student to better learn from the teacher distribution.

Kullback–Leibler (KL) distillation. Recall that we denote the teacher distribution by p and the student distribution by q_θ . Using the common KL divergence leads to the standard distilling objective

$$J_{\text{KL}} = D_{\text{KL}}(p||q_\theta) = \mathbb{E}_{\mathbf{Y} \sim p} \left[\log \frac{p(\mathbf{Y})}{q_\theta(\mathbf{Y})} \right] \quad (5)$$

$$\approx - \sum_{t=1}^{|\mathbf{y}|} \sum_{Y_t \in V} p(Y_t|\mathbf{y}_{<t}) \log q_\theta(Y_t|\mathbf{y}_{<t}) + \text{const} \quad (6)$$

where \mathbf{y} is sampled² from the teacher distribution p . Here, the constant is the entropy of p , which can be ignored as it does not involve the student parameters.

Similar to SeqKD, such KL distillation may also suffer from the mode-averaging problem and learn

²In our method, the expectation (5) is approximated by one Monte Carlo-sampled sequence. We denote a sampled sequence by a lower letter \mathbf{y} .

an overly smooth distribution, because q_θ is in the denominator in (5).

However, our KL distillation differs from SeqKD in that we adopt soft labels from the teacher model, i.e., keeping the entire distribution of $p(Y_t|\mathbf{y}_{<t})$, whereas SeqKD uses a certain decoded sequence \mathbf{y} as shown in Eqn. (3). Experiments will show that our soft labels provide more information than hard SeqKD in sequence-level distilling tasks, which is consistent with early evidence (Buciluă et al., 2006; Hinton et al., 2015).

Reverse KL (RKL) distillation. We propose RKL distillation, which can potentially address the mode-averaging problem:

$$\begin{aligned} J_{\text{RKL}} &= D_{\text{KL}}(q_\theta||p) = \mathbb{E}_{\mathbf{Y}' \sim q_\theta} \left[\log \frac{q_\theta(\mathbf{Y}')}{p(\mathbf{Y}')} \right] \\ &\approx \sum_{t=1}^{|\mathbf{y}'|} \sum_{Y'_t \in V} \left[q_\theta(Y'_t|\mathbf{y}'_{<t}) \log q_\theta(Y'_t|\mathbf{y}'_{<t}) \right. \\ &\quad \left. - q_\theta(Y'_t|\mathbf{y}'_{<t}) \log p(Y'_t|\mathbf{y}'_{<t}) \right] \end{aligned} \quad (7)$$

where \mathbf{y}' is sampled from the student distribution. In other words, the loss can be decomposed into the negative log probability of the teacher’s predicted probability plus the entropy of the student.

RKL does not suffer from mode averaging because the student distribution q_θ goes to the numerator and does not have to cover the teacher distribution. Also, the entropy term in (7) penalizes the student for learning a wide-spreading distribution, further mitigating the mode-averaging problem.

However, RKL distillation has the opposite problem, known as mode collapsing, where the student only learns one or a few modes of the teacher distribution. This is because the RKL loss would be large, if $q_\theta(\mathbf{Y}')$ is high but $p(\mathbf{Y}')$ is low for some \mathbf{Y}' . As a result, the student tends to overly concentrate its probability mass on certain high-probability regions of the teacher model, which may not be ideal either (Figure 1b).

RKL distillation is related to the ENGINE distilling approach (Tu et al., 2020), which was originally designed to minimize the energy function defined by the teacher model. In particular, the ENGINE objective approximates RKL less the student entropy: $J_{\text{ENGINE}} = \mathbb{E}_{\mathbf{Y} \sim q_\theta} [-\log p(\mathbf{Y})]$. Therefore, ENGINE also suffers from the mode-collapsing problem, resembling RKL distillation.

Remarks. KL and RKL have the mode-averaging or mode-collapsing problem, because

$D_{\text{KL}}(\cdot|\cdot)$ is asymmetric in its two arguments, requiring the second distribution to cover the support of the first. In the following, we will propose two f -DISTILL variants based on symmetric divergence functions to seek a balance between these two extremes.

Jenson–Shannon (JS) distillation. Our proposed JS distillation minimizes the JS divergence, which measures the difference between two distributions and their average. We derive the step-wise decomposition of the sequence-level JS loss:

$$\begin{aligned} J_{\text{JS}} &= \frac{1}{2} \mathbb{E}_{\mathbf{Y} \sim p} \left[\log \frac{p(\mathbf{Y})}{m(\mathbf{Y})} \right] + \frac{1}{2} \mathbb{E}_{\mathbf{Y}' \sim q_\theta} \left[\log \frac{q_\theta(\mathbf{Y}')}{m(\mathbf{Y}')} \right] \\ &\approx \frac{1}{2} \sum_{t=1}^{|\mathbf{y}|} \sum_{\mathbf{Y}_t \in V} -p(\mathbf{Y}_t | \mathbf{y}_{<t}) \log(m(\mathbf{Y}_t | \mathbf{y}_{<t})) \\ &+ \frac{1}{2} \sum_{t=1}^{|\mathbf{y}'|} \sum_{\mathbf{Y}'_t \in V} [q_\theta(\mathbf{Y}'_t | \mathbf{y}'_{<t}) \log(q_\theta(\mathbf{Y}'_t | \mathbf{y}'_{<t})) \\ &- q_\theta(\mathbf{Y}'_t | \mathbf{y}'_{<t}) \log(m(\mathbf{Y}'_t | \mathbf{y}'_{<t}))] + \text{const} \quad (8) \end{aligned}$$

where \mathbf{y} and \mathbf{y}' are sampled from the teacher’s and student’s distributions, which are compared with their average $m(\cdot) = \frac{1}{2}p(\cdot) + \frac{1}{2}q_\theta(\cdot)$. Appendix A provides the proof of this decomposition, and Subsection 2.3 presents an efficient approximation by avoiding on-the-fly sampling from the teacher.

Total variation distance (TVD) distillation. Our f -DISTILL gives rise to another novel distilling variant based on the total variation distance

$$J_{\text{TVD}} = \frac{1}{2} \sum_{\mathbf{Y}} |q_\theta(\mathbf{Y}) - p(\mathbf{Y})| \quad (9)$$

Unlike JS divergence, TVD measures the ℓ^1 norm between two distributions, and therefore does not have the log operator, making the gradient more stable than JS distillation.

We would like to decompose the sequence-level TVD step by step due to the intractable summation over the sentence space. However, TVD decomposition is non-trivial, and we show in Appendix A that the sequence-level TVD is upper bounded by step-wise terms, being our objective to minimize:

$$\begin{aligned} J_{\text{TVD}} &= \frac{1}{2} \sum_{\mathbf{Y}} |q_\theta(\mathbf{Y}) - p(\mathbf{Y})| \\ &\leq \frac{1}{4} \mathbb{E}_{\mathbf{Y} \sim p} \left[\sum_{t=1}^{|\mathbf{Y}|} \sum_{\mathbf{Y}_t \in V} |q_\theta(\mathbf{Y}_t | \mathbf{Y}_{<t}) - p(\mathbf{Y}_t | \mathbf{Y}_{<t})| \right] \end{aligned}$$

$$\begin{aligned} &+ \frac{1}{4} \mathbb{E}_{\mathbf{Y}' \sim q_\theta} \left[\sum_{t=1}^{|\mathbf{Y}'|} \sum_{\mathbf{Y}'_t \in V} |q_\theta(\mathbf{Y}'_t | \mathbf{Y}'_{<t}) - p(\mathbf{Y}'_t | \mathbf{Y}'_{<t})| \right] \\ &\approx \frac{1}{4} \sum_{t=1}^{|\mathbf{y}|} \sum_{\mathbf{Y}_t \in V} |q_\theta(\mathbf{Y}_t | \mathbf{y}_{<t}) - p(\mathbf{Y}_t | \mathbf{y}_{<t})| \\ &+ \frac{1}{4} \sum_{t=1}^{|\mathbf{y}'|} \sum_{\mathbf{Y}'_t \in V} |q_\theta(\mathbf{Y}'_t | \mathbf{y}'_{<t}) - p(\mathbf{Y}'_t | \mathbf{y}'_{<t})| \quad (10) \end{aligned}$$

where \mathbf{y} and \mathbf{y}' are again sampled from the teacher and student models, respectively.

Summary. In this part, we have described our proposed f -DISTILL framework with four variants based on different f -divergence functions. We have also presented their step-wise decompositions, whose justification is summarized by the following theorem, proved in Appendix A.

Theorem 1. (a) *The sequence-level KL, RKL, and JS divergences can be decomposed exactly into step-wise terms.* (b) *The sequence-level TVD can be upper bounded by step-wise terms.*

2.3 Implementation Considerations

Efficient approximation. Symmetric distilling losses (i.e., JS and TVD) are slow to compute, because they require sampling from both teacher and student models during training.

We propose to mitigate this by offline sampling for the teacher model to improve training efficiency. Specifically, we obtain teacher samples, i.e., \mathbf{y} in Eqns. (8) and (10), beforehand and keep them fixed during training. This is feasible because the teacher model is unchanged and hence does not require multiple inferences, whereas the student model is continuously updated and thus requires inference in an online fashion. Experiments show that such a treatment significantly improves the training efficiency for both JS and TVD distillations.

Pre-distillation. We warm-start our student model with the techniques developed by Shleifer and Rush (2020), who combine MLE training, word-level KL, and hidden state matching. Such a pre-distilling process is crucial to our f -DISTILL method, because most variants (namely, RKL, JS, and TVD distillations) require sampling from a student, but a randomly initialized student model generates poor samples, making the distilling process less meaningful.

Notice that, for a fair comparison, all baseline models are built upon the same pre-distilling process. This further confirms that our f -DISTILL

is compatible with existing techniques and yields add-on performance gain (shown in Section 3.2).

3 Experiments

3.1 Settings

Datasets and metrics. We evaluated f -DISTILL on a wide range of text generation tasks.

- **DART.** The DART dataset (Nan et al., 2021) is a popular data-to-text generation benchmark, where samples consist of structured data records and their corresponding text descriptions. We report common string-matching metrics, BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and TER (Snover et al., 2006), as well as popular learned metrics, BERTScore (Zhang et al., 2019), MoverScore (Zhao et al., 2019), and BLEURT (Sellam et al., 2020).

- **XSum.** Extreme Summarization (XSum, Narayan et al., 2018) is a large-scale dataset consisting of BBC articles and their one-sentence summaries. We report ROUGE scores, the most widely used metrics for summarization (Lin, 2004).

- **WMT16 EN-RO.** This dataset contains parallel texts for English and Romanian, and is one of the commonly used machine translation datasets (Bojar et al., 2016). We extracted 100K samples from the original dataset, as the teacher performance is nearly saturated at this size. We report BLEU, chrF (Popović, 2015), and TER scores for the translation quality, following existing machine translation literature (Sennrich et al., 2016; Barrault et al., 2019).

- **Commonsense Dialogue.** The Commonsense Dialogue dataset (Zhou et al., 2021) consists of dialogue sessions that are grounded on social contexts. We evaluated the output quality by BLEU and BERTScore. We only report BLEU1 and BLEU2, as higher-order BLEU scores are known to be unreliable for dialogue evaluation (Liu et al., 2016).

Model architectures. We evaluated f -DISTILL using state-of-the-art teacher models for different tasks. We followed the encoder–decoder architecture and used BART (Lewis et al., 2020) as the teacher for DART and XSum. We used T5 (Raffel et al., 2020), another encoder–decoder model, for WMT16 EN-RO, as it excels at machine translation. For Commonsense Dialogue, we followed Zhang et al. (2020b) and used DialoGPT, a decoder-only model pretrained on massive dialogue data.

Our student models followed the teachers’ architectures, but we reduced the number of layers. In

our experiments, we generally set the total number of layers to be four; specifically, encoder–decoder models had three encoder layers and one decoder layer, following the suggestion of deep encoders and shallow decoders in Kasai et al. (2020). For XSum, we set both the encoder and decoder to be three layers to compensate for the larger dataset. Additional experimental details can be found in Appendix B.

3.2 Results and Analyses

Main results. Table 2 presents the main results of our f -DISTILL along with a number of competing methods in the four experiments.

We first trained a neural network without distillation. The network was identical to our student model in terms of the neural architecture and hyperparameters, but we trained it directly by maximum likelihood estimation (MLE) based on ground-truth target sequences. As seen, the non-distilling model performs significantly worse than distilling methods, which agrees with existing literature and justifies the need for knowledge distillation (Hinton et al., 2015; Tang et al., 2019; Jiao et al., 2020).

We pre-distilled our student model based on Shleifer and Rush (2020), a classic distilling approach that combines ground-truth training, word-level distillation, and intermediate-layer matching. Our f -DISTILL approach requires pre-distillation, because it provides a meaningful initialization of the student model, from which our f -DISTILL would generate samples during training. That being said, all our distilling methods were built on the same pre-distilling model, constituting a fair comparison. The results show that, although the pre-distilling approach outperforms ground-truth MLE training, it is generally worse than other distilling methods. This implies that our contribution is “orthogonal” to existing methods, and that our f -DISTILL provides an add-on performance improvement.

We further experimented with SeqKD (Kim and Rush, 2016) and ENGINE (Tu et al., 2020), two established distilling methods in the distribution-matching category (see Section 1). They learn from hard sequences rather than probabilities, and thus are hard approximations of our KL and RKL distillations, respectively (Section 2.1). As seen, our soft label-based methods consistently outperform SeqKD and ENGINE. This suggests that soft labels (i.e., probabilities) provide more informative super-

Model		DART					
		BLEU4 [↑]	METEOR [↑]	TER [↓]	BERTScore [↑]	MoverScore [↑]	BLEURT [↑]
Teacher		48.56	39.28	45.45	83.04	68.17	40.56
Student	Non-distill (MLE)	43.12	35.71	49.97	79.76	65.65	29.10
	Pre-distill	45.60	36.99	47.10	81.39	66.75	34.08
	SeqKD	45.54	37.17	47.49	81.15	66.65	32.88
	ENGINE	44.40	36.51	50.63	80.18	66.20	30.94
	KL	46.24	37.45	46.89	81.60	67.07	35.31
	RKL	45.63	37.35	47.91	81.41	67.02	35.08
	JS	<u>46.85</u>	<u>37.75</u>	<u>46.50</u>	<u>81.93</u>	<u>67.30</u>	<u>36.81</u>
TVD	46.95	37.88	46.35	82.08	67.36	37.17	

Model		XSum			WMT16 EN-RO			Commonsense Dialogue		
		ROUGE-1 [↑]	ROUGE-2 [↑]	ROUGE-L [↑]	BLEU4 [↑]	chrF [↑]	TER [↓]	BLEU1 [↑]	BLEU2 [↑]	BERTScore [↑]
Teacher		45.12	22.26	37.18	25.82	55.76	60.57	11.67	5.03	47.69
Student	Non-distill (MLE)	30.00	10.67	24.40	19.90	49.79	69.48	10.23	3.56	45.15
	Pre-distill	40.58	17.79	32.55	20.68	50.51	68.38	9.95	3.63	46.22
	SeqKD	39.13	17.53	32.34	21.20	50.81	67.66	10.85	4.17	46.94
	ENGINE	39.19	16.18	31.23	17.65	48.37	84.02	10.13	4.26	46.91
	KL	41.28	18.98	33.71	21.45	51.12	66.74	9.81	3.52	45.80
	RKL	<u>41.69</u>	19.02	33.92	20.46	50.33	70.78	10.48	4.01	46.68
	JS	41.65	<u>19.22</u>	<u>34.03</u>	21.91	51.5	<u>66.86</u>	11.55	4.83	47.61
TVD	41.76	19.30	34.10	21.73	51.13	66.94	11.39	4.73	47.30	

Table 2: Main results on the DART, XSum, WMT16 EN-RO, and Commonsense Dialogue (CD) datasets. The best student result is in **bold** and the second best is underlined. ^{↑/↓}The higher/lower, the better.

Dataset	DART		XSum		MT _{EN-RO}		CD	
TeacherDist	26.10		36.28		23.13		81.19	
Risk	R_{llh}	R_{cvg}	R_{llh}	R_{cvg}	R_{llh}	R_{cvg}	R_{llh}	R_{cvg}
KL	0.56	0.49	1.89	1.68	1.23	0.82	0.43	0.26
RKL	0.58	0.59	1.88	1.83	1.20	1.60	0.29	0.35
TVD	0.53	0.52	1.86	1.77	1.21	1.78	0.27	0.35
JS	0.51	0.48	1.88	1.75	1.13	1.34	0.30	0.33

Table 3: The likelihood risk R_{llh} and the coverage risk R_{cvg} for different f -DISTILL variants. A lower number indicates a higher likelihood or better coverage. We show the teacher diversity for each task by distinct bi-gram percentage (Li et al., 2016a) among five teacher-sampled outputs given a test input, which indicates the severity of multi-modality of a task.

vision signals than hard sentences for sequence-level distillation, which is consistent with early literature on classification tasks (Buciluă et al., 2006; Hinton et al., 2015).

Among our f -DISTILL variants, we further observe that symmetric distilling losses (JS and TVD) are consistently better than asymmetric ones (KL and RKL) across all datasets except for WMT16 EN-RO, where KL achieves a slightly better TER performance. A plausible reason is that the machine translation task is semantically grounded: given a source text, there are limited ways to translate, because the model output has to preserve the meaning of the input sentence. This is analogous to learning a uni-modal distribution, where mode averaging does not occur because there is only one mode. Despite this, JS and TVD perform better in all other scenarios, as their symmetric divergence can force the student to better learn from its teacher distribution. They rank first or second for all tasks in terms of most of the metrics in Table 2, consis-

tently and largely outperforming previous methods.

Likelihood and coverage. We further analyze the mode averaging and collapsing behaviors of different distilling methods in Table 3. We propose to measure these aspects by a likelihood risk R_{llh} and a coverage risk R_{cvg} .

The *likelihood risk* is computed by $R_{llh} = \frac{1}{|\mathcal{D}_{student}|} \sum_{\mathbf{y}' \in \mathcal{D}_{student}} -\log p(\mathbf{y}')$. Here, $\mathcal{D}_{student}$ is the set of sentences generated from the student, where we sample a sentence for each input in the test set; $p(\mathbf{y}')$ is the teacher’s predicted probability of a student-sampled sentence \mathbf{y}' . A large likelihood risk suggests that the student may have averaged the teacher’s modes, causing it to generate atypical sentences from the teacher’s point of view (Figure 1a).

On the contrary, the *coverage risk* is computed by $R_{cvg} = \frac{1}{|\mathcal{D}_{teacher}|} \sum_{\mathbf{y} \in \mathcal{D}_{teacher}} -\log q_{\theta}(\mathbf{y})$, where we use the student q_{θ} to evaluate a teacher-sampled sentence $\mathbf{y} \in \mathcal{D}_{teacher}$. This measures whether the teacher’s samples are typical from the student’s

point of view, i.e., how well a student covers the support of the teacher’s distribution. A large coverage risk means that the teacher’s typical outputs are not captured by the student, which is an indicator of mode collapse (Figure 1b).

In addition, we notice that mode averaging and collapsing are significantly affected by how “multi-modal” a task is. We propose to measure this by the distinct bi-gram percentage (Li et al., 2016a) of the teacher model (denoted by TeacherDist): for each test input, we sampled five outputs from the teacher and computed the percentage of distinct bi-grams, which is then averaged across the test set. As seen in Table 3, the dialogue task exhibits the highest diversity, i.e., it is the most multi-modal, whereas machine translation is the least multi-modal.

Comparing KL and RKL, we find that KL distillation consistently achieves lower R_{cvg} risks (i.e., better coverage) than RKL across all datasets. This confirms that KL distillation yields a smooth student distribution that covers the teacher’s, whereas RKL distillation does not have the covering property due to its mode-collapsing nature.

We further observe that RKL achieves significantly higher likelihood (given by a lower R_{llh}) on the Commonsense Dialogue dataset. This shows that the mode-collapsing phenomenon of RKL distillation allows the student to generate plausible responses for the one-to-many dialogue task (Figure 1b), whereas the mode-averaging KL distillation puts the student in some desolate area in the teacher’s distribution (Figure 1a). On the other hand, RKL does not achieve lower likelihood risks in other tasks, since their one-to-many phenomenon is not as severe as dialogue generation (Wei et al., 2019; Bao et al., 2020; Wen et al., 2023).

Referring back to Table 2, we see that mode-averaging KL distillation is preferred over RKL for less multi-modal tasks, such as machine translation (which has a low TeacherDist score), whereas mode-collapsing RKL is preferred for highly multi-modal tasks, such as dialogue generation (which has a higher TeacherDist score).

Last, our symmetric distilling objectives (JS and TVD) generally have moderate likelihood and coverage risks between the two extremes. This shows that they achieve a compromise between mode collapsing and averaging, allowing them to yield high performance in all tasks (Table 2).

Analysis of the student size. We analyze our f -DISTILL variants with different student sizes in

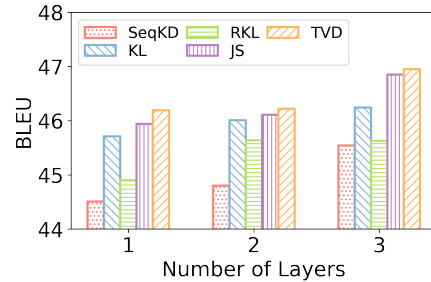


Figure 2: Comparison of KD methods when the student model has different numbers of encoder layers. Results were obtained on the DART dataset.

Model	BLEU4	BERTScore	Speedup
JS distillation			
Online	46.85	82.02	1.00x
Offline (our method)	46.85	81.93	2.25x
TVD distillation			
Online	46.57	82.03	1.00x
Offline (our method)	46.95	82.08	2.31x

Table 4: Training efficiency on the DART dataset. **Online:** We re-sample sequences from the teacher model in every epoch. **Offline:** The teacher’s samples are obtained beforehand and fixed during training. Note that we always re-sample from the student model because the student is constantly being updated.

comparison with the SeqKD model. Due to the limited time and resources, we chose the DART dataset as our testbed. We reduced the student model to different sizes by changing the number of encoder layers, as we had already used a single-layer decoder following the suggested architecture in Kasai et al. (2020). Results are shown in Figure 2.

As seen, our f -DISTILL outperforms SeqKD across all model sizes. The symmetric losses (JS and TVD) also consistently outperform the asymmetric ones (KL and RKL). This is consistent with our main results and further validates the effectiveness and robustness of our f -DISTILL framework.

Analysis of training efficiency. Our f -DISTILL involves sampling sequences from the teacher. We propose an offline approach that obtains the teacher’s samples before training. We analyze the efficiency of offline sampling for JS and TVD distillations by comparing them with their online counterparts. We ran this experiment on an NVidia RTX A6000 GPU and an Intel Xeon Gold 5317 CPU.³

As seen in Table 4, the offline variant achieves comparable performance, while the training speed is more than doubled. This is expected, as the of-

³To obtain a rigorous time estimate, we ran efficiency analysis on an unshared, consumer-grade server, whereas other experiments were run on clusters (Appendix B).

Model	Fluency [↑]	MissingInfo [↓]	Hallucination [↓]
SeqKD	4.75	1.77	1.67
ENGINE	4.51	1.76	1.61
JS	<u>4.72</u>	<u>1.70</u>	<u>1.48</u>
TVD	<u>4.72</u>	1.57	1.45

Table 5: Human evaluation on the DART dataset. Comparing SeqKD and TVD, the one-sided Student’s t -test gives p -values of 32.6%, 1.28%, and 0.669% for fluency, missing information, and hallucination, respectively.

fluent distilling methods do not require inference from the teacher model during training, which constitutes a significant portion of the training process. This shows that our symmetric distilling methods can achieve high performance without the need for sampling from both the teacher and student.

Human Evaluation. We further validated f -DISTILL by human evaluation, where models were rated by fluency, missing information, and hallucination between 1 to 5 on the DART dataset, following previous work (Nan et al., 2021; Keymanesh et al., 2022). We invited five human annotators to evaluate 50 test samples for four competing models: SeqKD, ENGINE, JS, and TVD. For each test sample, the annotators were presented with shuffled model outputs, so they could not tell which output was generated by which model. Results are shown in Table 5.

As seen, our f -DISTILL enables students to capture the input data records more faithfully while also retaining a high level of fluency. This is additionally supported by the p -values: comparing SeqKD and TVD, there is no statistically significant difference in terms of fluency (p -value=32.6%); however, the improvements for missing information (p -value=1.28%) and hallucination (p -value=0.669%) are statistically significant. Our human evaluation confirms the effectiveness of f -DISTILL.

Case Study. Appendix C shows example outputs for our f -DISTILL variants. Indeed, we observe KL distillation yields short and generic utterances that are believed to be an indicator of mode averaging (Wei et al., 2019; Bao et al., 2020). Our symmetric losses (JS and TVD) are able to generate more meaningful, fluent, and coherent sentences.

4 Related Work

Knowledge distillation (KD) is pioneered by Buciluă et al. (2006), who use an ensemble model as the teacher to train a single-model student by

minimizing the squared difference between their predicted logits. Hinton et al. (2015) propose to directly learn from the output probabilities by minimizing their KL divergence. Sun et al. (2019) propose patient knowledge distillation (PKD), which requires the student to learn from the teacher’s intermediate layers. Jiao et al. (2020) propose TinyBERT, extending knowledge distillation for Transformer models by additional treatments on the attention layers. Other recent distilling methods include finding the optimal layer mapping between two models (Li et al., 2020; Jiao et al., 2021) and learning from multiple teachers (Yang et al., 2020; Wu et al., 2021; Li et al., 2022).

The success of KD has since sparked significant interest in its applications to text generation. Kim and Rush (2016) investigate sequence-level knowledge distillation (SeqKD) for neural machine translation, where they use sampled, hard sequences to approximate the KL divergence. Tu et al. (2020) train a student model by minimizing the energy function defined by a teacher model, which we show is an approximation to reverse KL distillation. Lin et al. (2020) propose imitation-based KD, where the teacher provides oracle probabilities on student-sampled partial sequences to address the exposure bias problem. Further, KD has been extensively used to train non-autoregressive text generation models to reduce the complexity of the training data (Gu et al., 2018; Shao et al., 2022; Huang et al., 2022).

It is noted that our f -DISTILL requires meaningful student sampling and thus is built upon existing KD techniques (Shleifer and Rush, 2020), including word-level and intermediate-layer KD. Nevertheless, it shows that our approach achieves an add-on performance improvement, and that our contributions are orthogonal to previous work.

Besides KD, common model compression techniques include parameter pruning and sparse modeling. Parameter pruning first trains a dense network and then removes certain neural weights in hopes of not significantly affecting the model performance (LeCun et al., 1989; Liu et al., 2018; Fan et al., 2021). Alternatively, one may apply sparse modeling techniques such as regularization during the training process to ensure zero-valued parameters (Frankle and Carbin, 2018; Louizos et al., 2018; Tang et al., 2022). Our work does not follow these directions, as we consider the knowledge distilling setting.

Regarding the f -divergence function, it has many applications in the machine learning literature. The standard cross-entropy training is equivalent to minimizing the KL divergence between the ground-truth label distribution (often one-hot) and model distribution (Bishop, 2006). Generative adversarial networks (Goodfellow et al., 2014) minimize the Jensen–Shannon divergence by simultaneously training a generator and a discriminator against each other. Zhao et al. (2020) minimize α -divergence for adversarial learning, which generalizes KL and RKL, and is a special case of f -divergence functions. Zhang et al. (2021) use total variation distance as a regularizer to encourage the model to predict more distinguishable probabilities. Further, JSD is used in computer vision KD (Yin et al., 2020; Fang et al., 2021), but their tasks do not involve sequential data and the underlying techniques largely differ from our approach. To the best of our knowledge, we are the first to systematically formulate sequence-level knowledge distillation as f -divergence minimization.

5 Conclusion

We propose f -DISTILL, a family of sequence-level distilling methods beyond minimizing the KL divergence. Under our framework, we propose and analyze four variants: KL, RKL, JS, and TVD distillations, where existing SeqKD and ENGINE are approximations of KL and RKL variants; we further derive step-wise decomposition for our f -DISTILL. Results on four text generation tasks show f -DISTILL consistently outperforms existing KD methods, and that our symmetric losses (JS and TVD) outperform asymmetric ones by avoiding extreme mode averaging and collapsing.

6 Limitations

Our f -DISTILL variants are less efficient to train than SeqKD and ENGINE, as we require the teacher’s soft probabilities instead of hard, sampled sequences. However, our methods achieve a significant performance improvement, and more importantly, the additional training time does not affect inference when the model is deployed. This follows the spirit of knowledge distillation in general, i.e., to obtain a small and efficient model for deployment.

Another potential threat to validity is that we have not reported multi-run statistics. In our preliminary experiments, we ran our approach multiple

times and found results were generally consistent. Due to our excessive experimentation (estimated at 2000 GPU hours), it is not possible to run each model multiple times. We instead adopted a wide range of established automatic metrics, consistently showing the effectiveness of our approach. We further conducted in-depth analyses to better understand our proposed framework. We deem multi-run statistics not crucial to this paper, as this paper does not purely focus on empirical analysis. Rather, our main contributions lie in the novel machine learning framework, f -DISTILL, and the theoretical connections between step-wise and sequence-level f -divergence functions.

Acknowledgments

We thank all reviewers and chairs for their valuable comments. The research is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. RGPIN2020-04465, the Amii Fellow Program, the Canada CIFAR AI Chair Program, a UAHJIC project, a donation from DeepMind, and the Digital Research Alliance of Canada (alliancecan.ca).

References

- S. M. Ali and S. D. Silvey. 1966. [A general class of coefficients of divergence of one distribution from another](#). *Journal of the Royal Statistical Society*, 28(1):131–142.
- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An automatic metric for MT evaluation with improved correlation with human judgments](#). In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.
- Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. [PLATO: Pre-trained dialogue generation model with discrete latent variable](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 85–96.
- Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. [Findings of the 2019 Conference on Machine Translation \(WMT19\)](#). In *Proceedings of the Conference on Machine Translation*, pages 1–61.
- Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéal, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. [Findings of the 2016 Conference on Machine Translation](#). In *Proceedings of the Conference on Machine Translation*, pages 131–198.
- Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. [Model compression](#). In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 535–541.
- Chun Fan, Jiwei Li, Tianwei Zhang, Xiang Ao, Fei Wu, Yuxian Meng, and Xiaofei Sun. 2021. [Layer-wise model pruning based on mutual information](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 3079–3090.
- Gongfan Fang, Yifan Bao, Jie Song, Xinchao Wang, Donglin Xie, Chengchao Shen, and Mingli Song. 2021. [Mosaicking to distill: Knowledge distillation from out-of-domain data](#). In *Advances in Neural Information Processing Systems*, pages 11920–11932.
- Jonathan Frankle and Michael Carbin. 2018. [The lottery ticket hypothesis: Finding sparse, trainable neural networks](#). In *International Conference on Learning Representations*.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. [Generative adversarial nets](#). In *Advances in Neural Information Processing Systems*.
- Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. [Non-autoregressive neural machine translation](#). In *International Conference on Learning Representations*.
- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. [Distilling the knowledge in a neural network](#). *arXiv preprint arXiv:1503.02531*.
- Chenyang Huang, Hao Zhou, Osmar R. Zaiane, Lili Mou, and Lei Li. 2022. [Non-autoregressive translation with layer-wise prediction and deep supervision](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10776–10784.
- Xiaoqi Jiao, Huating Chang, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2021. [Improving task-agnostic BERT distillation with layer mapping search](#). *Neurocomputing*, 461:194–203.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. [TinyBERT: Distilling BERT for natural language understanding](#). In *Findings of the Association for Computational Linguistics: EMNLP*, pages 4163–4174.
- Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2020. [Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation](#). In *International Conference on Learning Representations*.
- Moniba Keymanesh, Adrian Benton, and Mark Dredze. 2022. [What makes data-to-text generation hard for pretrained language models?](#) In *Proceedings of the Workshop on Natural Language Generation, Evaluation, and Metrics*, pages 539–554.
- Yoon Kim and Alexander M. Rush. 2016. [Sequence-level knowledge distillation](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *International Conference on Learning Representations*.
- Rémi Lebreton, David Grangier, and Michael Auli. 2016. [Neural text generation from structured data with application to the biography domain](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213.
- Yann LeCun, John Denker, and Sara Solla. 1989. [Optimal brain damage](#). In *Advances in Neural Information Processing Systems*, pages 598–605.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.
- Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruifeng Xu, Min Yang, and Yaohong Jin. 2020. [BERT-EMD: Many-to-many layer mapping for BERT compression with earth mover’s distance](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 3009–3018.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. [A diversity-promoting objective function for neural conversation models](#). In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.
- Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. [Deep reinforcement learning for dialogue generation](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 4582–4597.

- Zhuoran Li, Chunming Hu, Xiaohui Guo, Junfan Chen, Wenyi Qin, and Richong Zhang. 2022. [An unsupervised multiple-task and multiple-teacher model for cross-lingual named entity recognition](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 170–179.
- Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020. [Autoregressive knowledge distillation through imitation learning](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 6121–6133.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81.
- Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. [How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.
- Liyuan Liu, Xiang Ren, Jingbo Shang, Xiaotao Gu, Jian Peng, and Jiawei Han. 2018. [Efficient contextualized representation: Language model pruning for sequence labeling](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1215–1225.
- Christos Louizos, Max Welling, and Diederik P Kingma. 2018. [Learning sparse neural networks through \$L_0\$ regularization](#). In *International Conference on Learning Representations*.
- Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. [DART: Open-domain structured data record to text generation](#). In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. [Don’t give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [BLEU: A method for automatic evaluation of machine translation](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 311–318.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2018. [A deep reinforced model for abstractive summarization](#). In *International Conference on Learning Representations*.
- Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation](#). In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. [Exploring the limits of transfer learning with a unified text-to-text Transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Igal Sason and Sergio Verdú. 2016. [f-divergence inequalities](#). *IEEE Transactions on Information Theory*, 62(11):5973–6006.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. [BLEURT: Learning robust metrics for text generation](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.
- Chenze Shao, Xuanfu Wu, and Yang Feng. 2022. [One reference is not enough: Diverse distillation with reference selection for non-autoregressive translation](#). In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3779–3791.
- Noam Shazeer and Mitchell Stern. 2018. [Adafactor: Adaptive learning rates with sublinear memory cost](#). In *Proceedings of the International Conference on Machine Learning*, pages 4596–4604.
- Sam Shleifer and Alexander M Rush. 2020. [Pre-trained summarization distillation](#). *arXiv preprint arXiv:2010.13002*.
- Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. [A study of translation edit rate with targeted human annotation](#). In *Proceedings of the Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. [Patient knowledge distillation for BERT model compression](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 4323–4332.

- Chuanxin Tang, Yucheng Zhao, Guangting Wang, Chong Luo, Wenxuan Xie, and Wenjun Zeng. 2022. [Sparse MLP for image recognition: Is self-attention really necessary?](#) In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2344–2351.
- Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. [Distilling task-specific knowledge from BERT into simple neural networks](#). *arXiv preprint arXiv:1903.12136*.
- Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. 2020. [ENGINE: Energy-based inference networks for non-autoregressive machine translation](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 2819–2826.
- Bolin Wei, Shuai Lu, Lili Mou, Hao Zhou, Pascal Poupart, Ge Li, and Zhi Jin. 2019. [Why do neural dialog systems generate short and meaningless replies? A comparison between dialog and translation](#). In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 7290–7294.
- Yuqiao Wen, Yongchang Hao, Yanshuai Cao, and Lili Mou. 2023. [An equal-size hard EM algorithm for diverse dialogue generation](#). In *International Conference on Learning Representations*.
- Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2021. [One teacher is enough? Pre-trained language model distillation from multiple teachers](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, pages 4408–4413.
- Ze Yang, Linjun Shou, Ming Gong, Wutao Lin, and Daxin Jiang. 2020. [Model compression with two-stage multi-teacher knowledge distillation for web question answering system](#). In *Proceedings of the International Conference on Web Search and Data Mining*, page 690–698.
- Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. 2020. [Dreaming to distill: Data-free knowledge transfer via DeepInversion](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. [PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization](#). In *Proceedings of the International Conference on Machine Learning*, pages 11328–11339.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. [BERTScore: Evaluating text generation with BERT](#). In *International Conference on Learning Representations*.
- Yivan Zhang, Gang Niu, and Masashi Sugiyama. 2021. [Learning noise transition matrix from only noisy labels via total variation regularization](#). In *Proceedings of the International Conference on Machine Learning*, pages 12501–12512.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. [DIALOGPT : Large-scale generative pre-training for conversational response generation](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.
- Miaoyun Zhao, Yulai Cong, Shuyang Dai, and Lawrence Carin. 2020. [Bridging maximum likelihood and adversarial learning via \$\alpha\$ -divergence](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6901–6908.
- Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. [MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 563–578.
- Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2021. [Commonsense-focused dialogues for response generation: An empirical study](#). In *Proceedings of the Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 121–132.

A Proof of Theorem 1

Theorem 1. (a) The sequence-level KL, RKL, and JS divergences can be decomposed exactly into step-wise terms. (b) The sequence-level TVD can be upper bounded by step-wise terms.

Proof. [**Part (a)**] We first consider the JS decomposition. Let p and q_θ be the predicted distribution for the teacher and student, respectively. Let $m(\mathbf{Y}) = \frac{1}{2}p(\mathbf{Y}) + \frac{1}{2}q_\theta(\mathbf{Y})$ be their average. We claim that JS divergence between two length- T sequence⁴ distributions can be decomposed step by step as

$$D_{\text{JS}}(p(\mathbf{Y}_{1:T}) \| q_\theta(\mathbf{Y}_{1:T})) := \frac{1}{2} \mathbb{E}_{\mathbf{Y}_{1:T} \sim p} \left[\log \frac{p(\mathbf{Y}_{1:T})}{m(\mathbf{Y}_{1:T})} \right] + \frac{1}{2} \mathbb{E}_{\mathbf{Y}'_{1:T} \sim q_\theta} \left[\log \frac{q_\theta(\mathbf{Y}'_{1:T})}{m(\mathbf{Y}'_{1:T})} \right] \quad (11)$$

$$\begin{aligned} &= \frac{1}{2} \sum_{t=1}^T \mathbb{E}_{\mathbf{Y}_{1:t-1} \sim p} \left[\sum_{Y_t} p(Y_t | \mathbf{Y}_{1:t-1}) \log \frac{p(Y_t | \mathbf{Y}_{1:t-1})}{m(Y_t | \mathbf{Y}_{1:t-1})} \right] \\ &+ \frac{1}{2} \sum_{t=1}^T \mathbb{E}_{\mathbf{Y}'_{1:t-1} \sim q_\theta} \left[\sum_{Y'_t} q_\theta(Y'_t | \mathbf{Y}'_{1:t-1}) \log \frac{q_\theta(Y'_t | \mathbf{Y}'_{1:t-1})}{m(Y'_t | \mathbf{Y}'_{1:t-1})} \right] \end{aligned} \quad (12)$$

For implementation, we use Monte Carlo (MC) sampling to approximate $\mathbb{E}_{\mathbf{Y}_{1:t-1} \sim p}[\cdot]$ and $\mathbb{E}_{\mathbf{Y}'_{1:t-1} \sim q_\theta}[\cdot]$, suggested by Eqn. (8). Then, we explicitly enumerate all Y_t and Y'_t , because a summation over all sequences is not tractable but a step-by-step summation over words is tractable. Compared with a direct MC approximation for (11), such step-wise decomposition allows us to propagate gradient into all the words (denoted by Y_t for the teacher and Y'_t for the student) for every step t .

In fact, the partially sampled sequences are reused for the summation over $t = 1, \dots, T$. That is to say, we will first sample the sequences $\mathbf{y}_{1:T-1} \sim p$ and $\mathbf{y}'_{1:T-1} \sim q_\theta$ and then compute the summation; thus, the complexity is linear rather than quadratic.

To prove (12), we first focus on the first term of (11):

$$\mathbb{E}_{\mathbf{Y}_{1:T} \sim p} \left[\log \frac{p(\mathbf{Y}_{1:T})}{m(\mathbf{Y}_{1:T})} \right] \quad (13)$$

$$= \mathbb{E}_{\mathbf{Y}_{1:T} \sim p} \left[\log \prod_{t=1}^T \frac{p(Y_t | \mathbf{Y}_{1:t-1})}{m(Y_t | \mathbf{Y}_{1:t-1})} \right] \quad (14)$$

$$= \mathbb{E}_{\mathbf{Y}_{1:T} \sim p} \left[\log \prod_{t=1}^{T-1} \frac{p(Y_t | \mathbf{Y}_{1:t-1})}{m(Y_t | \mathbf{Y}_{1:t-1})} + \log \frac{p(Y_T | \mathbf{Y}_{1:T-1})}{m(Y_T | \mathbf{Y}_{1:T-1})} \right] \quad (15)$$

$$= \mathbb{E}_{\mathbf{Y}_{1:T} \sim p} \left[\log \prod_{t=1}^{T-1} \frac{p(Y_t | \mathbf{Y}_{1:t-1})}{m(Y_t | \mathbf{Y}_{1:t-1})} \right] + \mathbb{E}_{\mathbf{Y}_{1:T} \sim p} \left[\log \frac{p(Y_T | \mathbf{Y}_{1:T-1})}{m(Y_T | \mathbf{Y}_{1:T-1})} \right] \quad (16)$$

$$= \mathbb{E}_{\mathbf{Y}_{1:T-1} \sim p} \left[\log \frac{p(\mathbf{Y}_{1:T-1})}{m(\mathbf{Y}_{1:T-1})} \right] + \mathbb{E}_{\mathbf{Y}_{1:T-1} \sim p} \left[\sum_{Y_T} p(Y_T | \mathbf{Y}_{1:T-1}) \log \frac{p(Y_T | \mathbf{Y}_{1:T-1})}{m(Y_T | \mathbf{Y}_{1:T-1})} \right] \quad (17)$$

where (14) decomposes $p(\mathbf{Y}_{1:T})$ and $m(\mathbf{Y}_{1:T})$; (15) and (16) split the T th step out. In (17), the first term drops Y_T because it does not occur in the expectation, and we rewrite the second term by making the summation over Y_t explicit in accordance with our sampling procedure.

Then, we can unroll the first term of (17) recursively, resulting in

$$\mathbb{E}_{\mathbf{Y}_{1:T} \sim p} \left[\log \frac{p(\mathbf{Y}_{1:T})}{m(\mathbf{Y}_{1:T})} \right] = \sum_{t=1}^T \mathbb{E}_{\mathbf{Y}_{1:t-1} \sim p} \left[\sum_{Y_t} p(Y_t | \mathbf{Y}_{1:t-1}) \log \frac{p(Y_t | \mathbf{Y}_{1:t-1})}{m(Y_t | \mathbf{Y}_{1:t-1})} \right] \quad (18)$$

⁴In practice, T can be thought of as the maximum length. Alternatively, we may consider varying-length sequences by a mixture of different values of T .

Likewise, the term $\mathbb{E}_{\mathbf{Y}'_{1:T} \sim q_\theta} \left[\log \frac{q_\theta(\mathbf{Y}'_{1:T})}{m(\mathbf{Y}'_{1:T})} \right]$ in (11) is treated in a similar fashion, concluding our proof for JS decomposition.

We state KL and RKL decompositions below. Their proofs are similar and thus omitted.

$$D_{\text{KL}}(p(\mathbf{Y}_{1:T}) \| q_\theta(\mathbf{Y}_{1:T})) = \sum_{t=1}^T \mathbb{E}_{\mathbf{Y}_{1:t-1} \sim p} \left[\sum_{Y_t} p(Y_t | \mathbf{Y}_{1:t-1}) \log \frac{p(Y_t | \mathbf{Y}_{1:t-1})}{q_\theta(Y_t | \mathbf{Y}_{1:t-1})} \right] \quad (19)$$

$$D_{\text{RKL}}(p(\mathbf{Y}_{1:T}) \| q_\theta(\mathbf{Y}_{1:T})) = \sum_{t=1}^T \mathbb{E}_{\mathbf{Y}'_{1:t-1} \sim q_\theta} \left[\sum_{Y'_t} q_\theta(Y'_t | \mathbf{Y}'_{1:t-1}) \log \frac{q_\theta(Y'_t | \mathbf{Y}'_{1:t-1})}{p(Y'_t | \mathbf{Y}'_{1:t-1})} \right] \quad (20)$$

[Part (b)] This part shows that the same step-wise decomposition for TVD is an upper bound:

$$D_{\text{TVD}}(p(\mathbf{Y}_{1:T}) \| q_\theta(\mathbf{Y}_{1:T})) := \frac{1}{2} \sum_{\mathbf{Y}_{1:T}} |q_\theta(\mathbf{Y}_{1:T}) - p(\mathbf{Y}_{1:T})| \quad (21)$$

$$\leq \frac{1}{2} \left[\frac{1}{2} \sum_{t=1}^T \mathbb{E}_{\mathbf{Y}_{1:t-1} \sim p} \left[\sum_{Y_t} |q_\theta(Y_t | \mathbf{Y}_{1:t-1}) - p(Y_t | \mathbf{Y}_{1:t-1})| \right] \right] \quad (22)$$

$$+ \frac{1}{2} \sum_{t=1}^T \mathbb{E}_{\mathbf{Y}'_{1:t-1} \sim q_\theta} \left[\sum_{Y'_t} |q_\theta(Y'_t | \mathbf{Y}'_{1:t-1}) - p(Y'_t | \mathbf{Y}'_{1:t-1})| \right] \quad (23)$$

We again start by re-writing the TVD loss in a recursive form

$$D_{\text{TVD}}(p(\mathbf{Y}_{1:T}) \| q_\theta(\mathbf{Y}_{1:T})) = \frac{1}{2} \sum_{\mathbf{Y}_{1:T}} |q_\theta(\mathbf{Y}_{1:T}) - p(\mathbf{Y}_{1:T})| \quad (24)$$

$$= \frac{1}{2} \sum_{\mathbf{Y}_{1:T-1}} \sum_{Y_T} |q_\theta(\mathbf{Y}_{1:T-1})q_\theta(Y_T | \mathbf{Y}_{1:T-1}) - p(\mathbf{Y}_{1:T-1})p(Y_T | \mathbf{Y}_{1:T-1})| \quad (25)$$

$$= \frac{1}{2} \sum_{\mathbf{Y}_{1:T-1}} \sum_{Y_T} \frac{p(\mathbf{Y}_{1:T-1})}{p(\mathbf{Y}_{1:T-1})} |q_\theta(\mathbf{Y}_{1:T-1})q_\theta(Y_T | \mathbf{Y}_{1:T-1}) - p(\mathbf{Y}_{1:T-1})p(Y_T | \mathbf{Y}_{1:T-1})| \quad (26)$$

$$= \frac{1}{2} \sum_{\mathbf{Y}_{1:T-1}} p(\mathbf{Y}_{1:T-1}) \sum_{Y_T} \left| \frac{q_\theta(\mathbf{Y}_{1:T-1})q_\theta(Y_T | \mathbf{Y}_{1:T-1})}{p(\mathbf{Y}_{1:T-1})} - p(Y_T | \mathbf{Y}_{1:T-1}) \right| \quad (27)$$

$$= \frac{1}{2} \mathbb{E}_{\mathbf{Y}_{1:T-1} \sim p} \left[\sum_{Y_T} \left| \frac{q_\theta(\mathbf{Y}_{1:T-1})q_\theta(Y_T | \mathbf{Y}_{1:T-1})}{p(\mathbf{Y}_{1:T-1})} - q_\theta(Y_T | \mathbf{Y}_{1:T-1}) + q_\theta(Y_T | \mathbf{Y}_{1:T-1}) - p(Y_T | \mathbf{Y}_{1:T-1}) \right| \right] \quad (28)$$

$$= \frac{1}{2} \mathbb{E}_{\mathbf{Y}_{1:T-1} \sim p} \left[\sum_{Y_T} \left| \frac{q_\theta(Y_T | \mathbf{Y}_{1:T-1})}{p(\mathbf{Y}_{1:T-1})} (q_\theta(\mathbf{Y}_{1:T-1}) - p(\mathbf{Y}_{1:T-1})) + q_\theta(Y_T | \mathbf{Y}_{1:T-1}) - p(Y_T | \mathbf{Y}_{1:T-1}) \right| \right] \quad (29)$$

$$\leq \frac{1}{2} \mathbb{E}_{\mathbf{Y}_{1:T-1} \sim p} \left[\sum_{Y_T} \left(\left| \frac{q_\theta(Y_T | \mathbf{Y}_{1:T-1})}{p(\mathbf{Y}_{1:T-1})} (q_\theta(\mathbf{Y}_{1:T-1}) - p(\mathbf{Y}_{1:T-1})) \right| + |q_\theta(Y_T | \mathbf{Y}_{1:T-1}) - p(Y_T | \mathbf{Y}_{1:T-1})| \right) \right] \quad (30)$$

$$= \frac{1}{2} \mathbb{E}_{\mathbf{Y}_{1:T-1} \sim p} \left[\sum_{Y_T} \left| \frac{q_\theta(Y_T | \mathbf{Y}_{1:T-1})}{p(\mathbf{Y}_{1:T-1})} (q_\theta(\mathbf{Y}_{1:T-1}) - p(\mathbf{Y}_{1:T-1})) \right| \right] +$$

$$\frac{1}{2} \mathbb{E}_{\mathbf{Y}_{1:T-1} \sim p} \left[\sum_{Y_T} |q_\theta(Y_T | \mathbf{Y}_{1:T-1}) - p(Y_T | \mathbf{Y}_{1:T-1})| \right] \quad (31)$$

$$= \frac{1}{2} \mathbb{E}_{\mathbf{Y}_{1:T-1} \sim p} \left[\left| \frac{1}{p(\mathbf{Y}_{1:T-1})} (q_\theta(\mathbf{Y}_{1:T-1}) - p(\mathbf{Y}_{1:T-1})) \right| \right] +$$

$$\frac{1}{2} \mathbb{E}_{\mathbf{Y}_{1:T-1} \sim p} \left[\sum_{Y_T} |q_\theta(Y_T | \mathbf{Y}_{1:T-1}) - p(Y_T | \mathbf{Y}_{1:T-1})| \right] \quad (32)$$

$$= \frac{1}{2} \sum_{\mathbf{Y}_{1:T-1}} |q_\theta(\mathbf{Y}_{1:T-1}) - p(\mathbf{Y}_{1:T-1})| + \frac{1}{2} \mathbb{E}_{\mathbf{Y}_{1:T-1} \sim p} \left[\sum_{Y_T} |q_\theta(Y_T | \mathbf{Y}_{1:T-1}) - p(Y_T | \mathbf{Y}_{1:T-1})| \right] \quad (33)$$

$$= \frac{1}{2} \sum_{t=1}^T \mathbb{E}_{\mathbf{Y}_{1:t-1} \sim p} \left[\sum_{Y_t} |q_\theta(Y_t | \mathbf{Y}_{1:t-1}) - p(Y_t | \mathbf{Y}_{<1:t})| \right] \quad (34)$$

where (25) breaks the sequence-level summation into the first $T - 1$ steps and the last step, (26) multiplies and divides $p(\mathbf{Y}_{1:T-1})$, and (28) subtracts and adds $q_\theta(Y_T | \mathbf{Y}_{1:T-1})$. After some regrouping in (29), we apply the triangle inequality in (30). In (31), we break the expectation into two terms, where the first term is further simplified by summing over Y_T in (32) and expanding the expectation in (33). These manipulations bring the equation to a recursive form. By applying the same technique as in (18), we may further unroll the first term in (33) and eventually obtain (34) as an upper bound.

Likewise, we can obtain the following inequality by multiplying and dividing by $q_\theta(y_{1:T-1})$ in (26)

$$\mathcal{L}_{\text{TVD}} \leq \frac{1}{2} \sum_{t=1}^T \mathbb{E}_{\mathbf{Y}'_{1:t-1} \sim q_\theta} \left[\sum_{Y'_t} |q_\theta(Y'_t | \mathbf{Y}'_{1:t-1}) - p(Y'_t | \mathbf{Y}'_{1:t-1})| \right] \quad (35)$$

These two upper bounds, (34) and (35), are then combined to obtain (23), concluding the proof. \square

Admittedly, both (34) and (35) are valid upper bounds for the TVD divergence, but we nevertheless combine these two formulas to obtain a more computationally robust upper bound in the same spirit of JS decomposition.

B Experimental Details

Table 6 shows the statistics of our datasets. As seen, we benchmarked our models on a variety of natural language generation tasks with different data sizes. We chose state-of-the-art models as the teachers, with 200M–400M parameters. Accordingly, our students had 50M–150M parameters. The high performance of f -DISTILL variants across these datasets highlights the robustness of our approach.

For training, we used the Adam optimizer (Kingma and Ba, 2015) with default hyperparameters $\beta = (0.9, 0.999)$ on DART, XSum, and Commonsense Dialogue. For WMT16 EN-RO, we followed the T5 teacher model (Raffel et al., 2020) and used the AdaFactor optimizer (Shazeer and Stern, 2018). We chose a small batch size of eight to fit the student as well as the large teacher in our GPU. All student models were trained for 28 epochs for pre-distillation and another 12 epochs for each distilling method, as additional training did not further improve performance.

Dataset	Task	# of Samples		
		Train	Dev	Test
DART (Nan et al., 2021)	Data-to-Text Generation	30,526	2,768	4,159
XSum (Narayan et al., 2018)	Summarization	204,045	11,332	11,334
WMT16 EN-RO (Bojar et al., 2016)	Machine Translation	100,000	1,999	1,999
Commonsense Dialogue (Zhou et al., 2021)	Dialogue Generation	51,831	6,619	6,610

Table 6: Statistics of our datasets.

DART	
Input	(11028/11027, destination, mumbai), (11028/11027, origin, Chennai), (11028/11027, train_name, mumbai Mail)
Reference	train no . 11028 / 11027 name of the train is mumbai mail has origin and destination of chennai to mumbai
KL	the train name of the train that originates from chennai is mumbai.
RKL	the mumbai mumbai mumbai train name is mumbai mumbai and the destination is mumbai.
JS	the mumbai mail train goes to mumbai and originates from chennai to mumbai.
TVD	the mumbai mail train starts from chennai to mumbai.
Commonsense Dialogue	
Input	Quinn spent many years studying. Finally it became graduation time for him.
Reference	I can't describe how happy I am on this day.
KL	I am done. Now is over.
RKL	I am going to miss my mom and dad. I am going to miss my dad and the people I have to do a lot of things.
JS	I am so excited for my graduation. I can't wait to get back to my home and spend some time with my family.
TVD	I am finally done packing my dorm. I can't wait to start my first year of college.

Table 7: Example outputs from f -DISTILL variants.

The main experiments were conducted on AMD Milan 7413 CPUs and NVidia A100 GPUs, and the total training time was estimated at 2000 GPU hours. Note that this is not because our algorithm is slow (efficiency analyzed in Table 4), but because we have extensively experimented with a variety of datasets and model variants.

C Case Study

Table 7 shows example outputs for DART and Commonsense Dialogue. On the DART dataset, the KL and RKL distillations fail to yield coherent responses from the input data records. By contrast, JS and TVD distillations enable the student to generate sentences of much higher quality: they correctly recognize the name of the train as well as its origin and destination.

We additionally show an example output from the Commonsense Dialogue dataset, because the dialogue task exhibits the most severe multi-modal problem, which in turn requires the student to carefully balance mode averaging and collapsing. As seen, the KL-distilled student generates a short and generic response, which is consistent with existing literature (Wei et al., 2019; Bao et al., 2020), explained as mode averaging in our paper. The RKL-distilled student generates a detailed, but ungrammatical and incoherent, response. For JS and TVD distillations, the students generate responses that are both coherent and detailed. The case studies confirm our main claim that JS and JVD are more effective sequence-level distilling approaches.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
The limitation section is right after the conclusion in Section 5.
- A2. Did you discuss any potential risks of your work?
Our work focuses on knowledge distillation for small models. It does not impose more risk than other machine learning/NLP research.
- A3. Do the abstract and introduction summarize the paper's main claims?
The main claims are summarized both in the Abstract and at the end of the Introduction (Section 1).
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

We published our code in Footnote 1 in the abstract.

- B1. Did you cite the creators of artifacts you used?
All the models and datasets used are cited in Section 3 of our paper.
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We did not distribute any artifacts other than our code base. The code base is provided as an anonymous GitHub repo and is under the MIT license.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We used and cited existing datasets and models, which are intended for research purposes. Our code is also intended for research purposes only.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We did not collect new datasets, and we made sure to use datasets that are well-established in the research community.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The documentation of our database will be provided through GitHub README.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
See Experimental Details in Appendix B.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C Did you run computational experiments?

Yes. Section 3.

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?

Yes, see Appendix B Experimental Details.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

Yes, see Appendix B Experimental Details.

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

Yes, we stated in Section 6 that multi-run variance was low in our preliminary experiments. Thus, main results were obtained by running the model only once, because we conducted extensive experimentation (four tasks and a large number of competing models) with various evaluation metrics. Running every model multiple times is neither necessary nor feasible.

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

We provide our entire code base as an anonymous GitHub repo in the abstract. The code base contains all details to reproduce our work.

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

No.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

Not applicable. Left blank.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

Not applicable. Left blank.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

Not applicable. Left blank.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

Not applicable. Left blank.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

Not applicable. Left blank.