Curriculum Consistency Learning for Conditional Sentence Generation

Liangxin Liu¹ Xuebo Liu^{1*} Lian Lian² Shengjun Cheng² Jun Rao¹ Tengfei Yu¹ Hexuan Deng¹ Min Zhang¹

¹Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen, China ²Huawei Cloud Computing Technologies Co., Ltd.

{lliangxin967,rao7jun,hxuandeng}@gmail.com, tengfeiyu@stu.hit.edu.cn

{lianlian3,chengshengjun}@huawei.com, {liuxuebo,zhangmin2021}@hit.edu.cn

Abstract

Consistency learning (CL) has proven to be a valuable technique for improving the robustness of models in conditional sentence generation (CSG) tasks by ensuring stable predictions across various input data forms. However, models augmented with CL often face challenges in optimizing consistency features, which can detract from their efficiency and effectiveness. To address these challenges, we introduce Curriculum Consistency Learning (CCL), a novel strategy that guides models to learn consistency in alignment with their current capacity to differentiate between features. CCL is designed around the inherent aspects of CL-related losses, promoting task independence and simplifying implementation. Implemented across four representative CSG tasks, including instruction tuning (IT) for large language models and machine translation (MT) in three modalities (text, speech, and vision), CCL demonstrates marked improvements. Specifically, it delivers +2.0 average accuracy point improvement compared with vanilla IT and an average increase of +0.7 in COMET scores over traditional CL methods in MT tasks. Our comprehensive analysis further indicates that models utilizing CCL are particularly adept at managing complex instances, showcasing the effectiveness and efficiency of CCL in improving CSG models. Code and scripts are available at https://github.com/xinxinxing/ Curriculum-Consistency-Learning.

1 Introduction

Consistency learning (CL) as a robust and widely utilized methodology for enhancing the performance of various conditional sentence generation (CSG) models, (Liu et al., 2023; Li et al., 2023b; Xie et al., 2022; Fang and Feng, 2023; Yin et al., 2023; Li et al., 2022b; Kambhatla et al., 2022; Li et al., 2022a). The primary goal of CL is to ensure uniformity in a model's predictions across identical or similar instances, even when faced with minor variations in data or model configurations. However, the integration of CL into models has faced challenges related to suboptimal convergence efficiency (Liang et al., 2021). The core issue arises from the increased computational demands required to optimize the CL loss, effectively doubling the convergence time for the models. Furthermore, with the ongoing growth in the complexity and size of models, the efficiency problem becomes increasingly significant (Hu et al., 2022).

This work is motivated by the need to enhance the learning efficiency of CSG models using CL. Our observations show that existing CL approaches use a uniform strategy to learn consistency features, which is suboptimal for CSG models. Early in training, models should prioritize developing strong feature representations over consistency learning. Once proficient in feature discrimination, models can then focus on consistency modeling. This phased approach aligns with curriculum learning principles (Bengio et al., 2009), which tailor the learning process to the model's evolving capabilities. However, integrating curriculum learning with CL methods is challenging. Curriculum learning depends on task-specific difficulty metrics and training schedules (Platanios et al., 2019; Soviany et al., 2020; Wang et al., 2021; Soviany et al., 2022), contrasting with the broad applicability of CL, adding complexity to this integration.

In this paper, we introduce a novel approach, Curriculum Consistency Learning (CCL), designed to overcome the existing limitations and broaden the applicability of CL in CSG models. CCL integrates a difficulty measure and a proficiency estimator directly derived from the CL losses observed in both the training and validation phases. This innovation eliminates the dependency on task-specific metrics, streamlining the implementation process. CCL strategically guides the model to engage in CL with instances that match its current proficiency

^{*}Co-first and Corresponding Author

level, ensuring more effective and efficient optimization of CL-enhanced CSG models, leading to improved performance.

Experimental evaluation in four CSG tasks, including instruction tuning (IT) tasks (Rao et al., 2024; Fan et al., 2024) for large language models (LLMs) and machine translation (MT) tasks in three distinct modalities (text, vision, and speech) establishes the superior performance of CCL over existing advanced baselines and conventional CL approaches. Specifically, it delivers +2.0 average accuracy point improvement compared with vanilla IT and an average increase of +0.7 in COMET scores over traditional CL methods in MT tasks. This advancement is supported by detailed analyses, which corroborate the effectiveness of CCL in improving the model's ability to handle complex instances, thus confirming the benefits of our method in enhancing the efficiency and efficacy of models in diverse CSG tasks.

Our contributions are summarized as follows:

- We introduce CCL, an innovative framework for the training of CSG models, which strategically enhances model consistency learning across differentiated training phases.
- We propose a novel automatic metric for the difficulty of instances and the capability of models within the context of the CL paradigm.
- We demonstrate that the notable performance improvements by CCL can be largely attributed to its superior handling of challenging samples, such as translating hard sentences.

2 Background and Existing Challenges

2.1 Consistency Learning

CL is a technique that seeks to establish uniformity between the predictions of the model from different views on the same sample. The fundamental concept revolves around creating cross-view supervision and enhancing cross-view consistency through the implementation of a CL loss, thereby augmenting supervision signals. In this paper, we primarily reference the exemplary CL method R-Drop (Liang et al., 2021) to elucidate the background. The overall loss function of a CL-enhanced model can be expressed as follows:

$$\mathcal{L} = \arg\min\sum_{n=1}^{N} \left(\underbrace{\sum_{i=1}^{I} -\log P(y^{n}|x_{i}^{n}, \theta_{i})}_{\text{Negative Log-likelihood Loss:}\mathcal{L}_{\text{NLL}}} + \underbrace{\alpha \text{Divergence}(\{P(y^{n}|x_{i}^{n}, \theta_{i})\}_{i=1}^{I})}_{\text{Consistency Learning Loss:}\mathcal{L}_{\text{CL}}} \right)$$
(1)

where I means the number of different views of the sample and $\mathcal{D} = \{\langle x^n, y^n \rangle\}_{n=1}^N$ denotes the training set. x_1^n and x_i^n respectively represent the default or variations version of x^n , θ_i means model parameters.

Training Set \mathcal{D} In the context of models, $\mathcal{D} = \{\langle x^n, y^n \rangle\}_{n=1}^N$ may represent the training set for diverse tasks. For example, both x and y can represent a sentence in IT or textual translation tasks.

Negative Log-likelihood Loss \mathcal{L}_{NLL} The Negative Log-likelihood (NLL) loss constitutes the aggregation of losses stemming from varied predictions across multiple views. These views (aka consistency sources) can be engendered through either x or θ . For instance, x_i may be a semantically equivalent version (Kambhatla et al., 2022) of x; alternatively, θ_i might represent the same model but with varying structures resulting from random dropout masks (Liang et al., 2021).

Consistency Learning Loss \mathcal{L}_{CL} The CL loss constitutes the crux of CL. Different divergence functions, such as Jensen–Shannon (JS) (Lin, 1991) and Kullback–Leibler (KL) divergences (Kullback and Leibler, 1951), can be employed based on specific context. For instance, consider the bidirectional KL divergence used in R-Drop:

$$\mathcal{L}_{CL}^{n} = \alpha \text{Divergence}(\{P(y^{n}|x^{n},\theta_{i})\}_{i=1}^{2})$$

$$= \alpha(\text{KL}(P(y^{n}|x^{n},\theta_{1})||P(y^{n}|x^{n},\theta_{2})) + \text{KL}(P(y^{n}|x^{n},\theta_{2})||P(y^{n}|x^{n},\theta_{1}))) \quad (2)$$

In this equation, α is a coefficient that balances the learning of the two losses of NLL Loss and CL Loss. While x^n remains constant in the two predictions, θ varies due to the disparate forward propagations induced by random dropout masks, yielding the different consistency sources $P(y^n | x^n, \theta_1)$ and $P(y^n | x^n, \theta_2)$.



Figure 1: The overall framework of CCL. In the CL-aware Difficulty Measurer, CCL utilizes the intrinsic consistency loss to evaluate the sample difficulty. In the Model Proficiency Estimator, the training and validation sets are used to jointly determine model proficiency. Based on the sample difficulty and model proficiency, CCL can calculate the sample weight for the model training.

Challenge The preceding discussion demonstrates the versatility of CL concerning tasks, consistency sources, and loss types. By minimizing the final loss function \mathcal{L} , CL can force the output distributions of different models or data to align, thereby effectively enhancing the model's generalization capabilities. However, the process of constraining the consistency distribution of the model, coupled with the inherent challenges of multi-objective optimization, often requires CL to invest more training time in achieving model convergence.

2.2 Curriculum Learning

To mitigate the above limitations, this paper primarily draws inspiration from the field of curriculum learning (Bengio et al., 2009). Curriculum learning commences by training on simpler tasks or more straightforward examples, gradually exposing the network to more complex tasks or challenging examples. This process significantly simplifies the learning problem by directing the learning algorithm towards better local minima within the optimization landscape. Consequently, this progressive approach can enable models to learn more effectively and potentially at a faster pace.

Challenge Despite its potential benefits, the application of curriculum learning presents several challenges. It demands the establishment of distinct difficulty metrics for tasks across various modalities to determine which instances are simple or hard. For instance, sentence length (Platanios et al., 2019) and word rarity (Kocmi and Bojar, 2017) are metrics used to gauge the difficulty of textual instances, which are not suitable for visual features. Moreover, it requires the determination of the appropriate time for the models to engage with easy or difficult instances (Platanios et al., 2019). This complexity poses significant obstacles to the integration of curriculum learning with CL.

3 Our Proposed Method

We propose that CL should not be the primary focus until the model has developed sufficient meaningful features and representations. Applying consistency constraints too early can lead to learning incorrect features and slow optimization. To address this, we introduce CCL, which adapts consistency learning to different training stages. As shown in Figure 1, we first measure instance difficulty using the consistency loss of the training set, eliminating the need to model different modalities separately. Next, we calculate the average CL loss for both the training and validation sets to gauge the model's capacity. Finally, we calculate the instance weights and update model parameters, ensuring the model focuses on tasks aligned with its strengths.

3.1 CL-aware Difficulty Measurer

As previously elaborated, when a model with limited capacity encounters complex instances, it struggles to capture intricate features and patterns within the data. This difficulty manifests as a significant increase in the value of the CL loss, rendering the learning process ineffective. To address this challenge, we introduce the use of CL loss as a measure of instance difficulty in this paper. This metric is both straightforward to acquire and cost-effective during the model training process. Importantly, it is determined intrinsically by the model, obviating the need for human intervention.

Formulation We first collected the CL loss \mathcal{L}_{CL}^n of all training instances for difficulty assessment:

$$S^{n} \in [0,1) = \frac{\mathcal{L}_{CL}^{n} - \min{\{\mathcal{L}_{CL}^{n}\}_{n=1}^{N}}}{\max{\{\mathcal{L}_{CL}^{n}\}_{n=1}^{N}}} \quad (3)$$

Where S^n represents the instance difficulty. The interpretation of this measure is straightforward: the larger the value of the CL loss, the more challenging the instance is for the model to learn consistency at that particular juncture.

3.2 Model Proficiency Estimator

To evaluate a model's ability to learn and model consistency, we propose a novel estimation method centered around the CL loss within the validation set and training set. We calculate the average CL losses on the training set (\mathcal{L}_{CL}^{Train} = average({ \mathcal{L}_{CL}^n }^N_{n=1}) and validation set (\mathcal{L}_{CL}^{Valid} = average({ \mathcal{L}_{CL}^m }^N_{n=1}) at each epoch. During the model training, we found that the CL loss in the training set remains stable, whereas the CL loss in the validation set increases as the training progresses. One possible reason is that the CL loss is typically much smaller than the model loss throughout the training process in standard CL methods. This discrepancy might arise because the primary optimization target is the model loss, while CL loss plays a supplementary role in aiding this optimization. As a result, the CL loss on the training set

remains relatively stable. However, as the model becomes more adept at capturing consistency due to the constraints of the CL loss, there's a noticeable increase in the CL loss on the validation set. So, this growing disparity, resulting from the constraints of consistency, can serve as an effective measure of model proficiency C in CL.

Formulation We quantify the capability of the model by employing \mathcal{L}_{CL}^{Train} and \mathcal{L}_{CL}^{Valid} as follows:

$$C = \min\left(1, \frac{\left(\mathcal{L}_{\rm CL}^{\rm Valid} - \mathcal{L}_{\rm CL}^{\rm Train}\right)}{T_{max}}\right) \qquad (4)$$

where T_{max} means the value of \mathcal{L}_{CL}^{Valid} - \mathcal{L}_{CL}^{Train} at 80% of the total training steps of vanilla CL. We follow Liu et al. (2020); Platanios et al. (2019) to stop our curriculum learning time at later of the training process to enable the model to perform unbiased learning in the later stages of training. A distinct advantage of our approach is its ability to encapsulate the model proficiency from a consistency perspective, and it is entirely data and model-driven, eliminating the necessity for human intervention.

3.3 Instance Weight Calculation

The proposed method facilitates the model in learning instances that align more precisely with its current ability. This is achieved by adjusting the weight of each instance w^n , increasing it when the difficulty of an instance matches the model's current proficiency, and decreasing it when the instance is either excessively challenging or too simple for the model. By varying the weights of different instances within the loss function, the model's focus and significance assigned to these instances can be accordingly manipulated, as detailed below:

$$w^{n} = exp(1 - |C - S^{n}|) - 1$$
 (5)

where the S^n represents the instance difficulty score and C is the model proficiency. When the difficulty of an instance closely matches the model proficiency, the model assigns it greater attention $(w^n > 1)$. Conversely, if the instance is too difficult or too easy for the model, it will be assigned less attention $(w^n < 1)$.

3.4 Model Update

Based on the analysis of model proficiency, we choose to apply the curriculum learning part w^n on the original loss function, rather than only the CL

ID System	Ext	ernal	MMLU	BBH	GSM	TydiQA	CodeX	AE	Ov	erall
ib System	Data	Model		DDII	00111	i yui Qii	couch		AVG	$\Delta\left(\uparrow ight)$
Base Model: LLaMA-2-7B		Imp	lemented E	xisting l	Method					
1 Alpaca-GPT4			46.2	39.2	15.0	43.3	27.8	33.7	34.3	-
2 1 + AlpaGasus	×	 Image: A second s	46.8	39.2	14.5	48.4	26.5	34.6	35.0	+0.8
3 1 + Q2Q	×	 Image: A second s	46.7	39.8	16.5	45.5	28.1	35.1	35.3	+1.1
4 1 + Instruction Mining	\checkmark	 Image: A second s	47.0	40.0	16.5	47.8	29.6	34.4	35.9	+1.7
5 1 + Data CL	×	×	47.3	39.0	14.5	43.7	29.5	35.1	29.5	+0.6
			Our Met	hod						
6 5 + CCL	×	×	47.6	40.8	15.5	48.4	30.7	35.6	36.4	+2.1
Base Model: LLaMA-2-13B		Imp	lemented E	xisting I	Method					
7 Alpaca-GPT4			55.7	47.3	31.0	49.1	41.8	46.5	45.2	-
8 7 + AlpaGasus	×	\checkmark	54.1	49.3	32.0	52.6	39.3	47.5	45.8	+0.6
9 7 + Q2Q	×	 Image: A second s	55.3	48.5	34.0	50.8	42.3	47.7	46.4	+1.2
10 7 + Instruction Mining	 Image: A second s	 Image: A second s	55.5	49.7	33.0	51.2	41.5	46.1	46.2	+1.0
11 7 + Data CL	×	×	55.2	47.2	33.0	51.2	40.8	46.2	45.6	+0.4
			Our Met	hod						
12 11 + CCL	×	×	55.6	49.3	34.0	53.6	42.7	47.6	47.1	+1.9

Table 1: The overall results on Instruction Tuning of LLMs. Codex and AE mean HumanEval and AlpacaEval benchmark respectively. CCL significantly enhances the IT performance of LLMs.

loss, which is much smaller and hard to produce a significant impact and change to the total loss (Liu et al., 2020; Platanios et al., 2019). By integrating w^n with the original loss function, the final loss function of CCL can be represented as follows:

$$\mathcal{L}^* = \arg\min\sum_{n=1}^{N} w^n (\mathcal{L}_{\mathrm{NLL}}^n + \alpha \mathcal{L}_{\mathrm{CL}}^n) \quad (6)$$

Model Generality It is important to note that both the instance difficulty and model proficiency estimation are derived from existing computational outcomes, necessitating minimal additional computations. Consequently, our proposed CCL methodology is not only straightforward to implement, but it also demonstrates a high level of generality, making it suitable for a variety of CSG scenarios.

4 Evaluation

To establish the universality of our approach, we evaluate it using both unimodal and multimodal tasks. We focus on the following four CSG tasks:

Instruction Tuning (IT) uses instruction data pairs to perform supervised fine-tuning on LLMs, which can effectively align LLMs with human preferences. We use the popular Alpaca-GPT4 dataset (Peng et al., 2023) and the LLaMA-2 model family (Touvron et al., 2023) as our training sets and foundation model respectively.

Textual Machine Translation (TMT) involves translating a sentence from one language to another while retaining semantic equivalence. We employ the vanilla Transformer model (Vaswani et al., 2017) and machine translation LLM ALMA-7B (Xu et al., 2024) as our baseline and test their performance on two prominent benchmarks IWSLT14 German-English benchmarks and WMT22 English-Chinese benchmarks respectively.

Multimodal Machine Translation (MMT) is a task aimed at using a fusion of text and image data to enhance the quality and expressive capacity of translation. We run experiments on the widelyused Multi30K English-German benchmark (Elliott et al., 2016), using VL-T5 (Cho et al., 2021) as our baseline due to its superior performance.

Speech-to-Text Translation (ST) translates source acoustic speech signals in a source language into a foreign text without any intermediate output. We use the commonly-used MUST-C dataset (Di Gangi et al., 2019) and focus on the English-German translation task. The SpeechUT model (Zhang et al., 2022), known for its strong performance, is chosen as our baseline.

4.1 Evaluation Metric

For the IT tasks, we comprehensively evaluate the LLMs on the open-instruct code repository (Wang et al., 2023), including MMLU (Hendrycks et al., 2021), GSM (Cobbe et al., 2021), BBH (Suzgun et al., 2022), TyDiQA (Clark et al., 2020), HumanEval (Chen et al., 2021b), and AlpacaEval (Dubois et al., 2023). For the MT tasks (TMT, MMT, and ST), we report both the widely used BLEU score (Post, 2018; Ott et al., 2018), which

		TMT				MMT				ST	
ID	Method	IWS	SLT14	WMT2	2 (ALMA)	Test16		Test17		MUST-C	
		BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET
				E	xisting Lead	ing Metho	od				
13	R-Drop	37.3	-	-	-	-	-	-	-	-	-
14	ALMA	-	-	36.5	85.1	-	-	-	-	-	-
15	VL-T5	-	-	-	-	45.3	-	42.4	-	-	-
16	SpeechUT	-	-	-	-	-	-	-	-	30.1	-
					Implemented	d Method					
17	Baseline	34.6	74.1	36.3	84.8	44.9	72.0	42.0	72.6	29.9	79.5
18	17 + Data CL	37.5	76.7	36.0	84.3	45.7	72.7	42.4	72.8	30.2	79.7
19	17 + Model CL	37.3	76.5	-	-	45.6	72.4	42.2	72.5	30.2	79.8
	Our Method										
20	18 + CCL	38.0 [‡]	77.2^{\dagger}	37.0 [‡]	85.2 [†]	46.1^{\dagger}	73.3 [‡]	42.9 [†]	73.5 †	30.6 [†]	80.3 [‡]
21	19 + CCL	37.8 [‡]	77.1^{+}	-	-	45.9^{\dagger}	72.9 [‡]	42.7^{\dagger}	73.4 [‡]	30.6 [‡]	80.4 [‡]

Table 2: The overall results on three MT tasks. "Baseline" are our re-implemented baseline, namely Transformer (Vaswani et al., 2017), ALMA (Xu et al., 2024), VL-T5 (Cho et al., 2021) and SpeechUT (Zhang et al., 2022). " \dagger " and " \ddagger " the improvement is significant by contrast to the corresponding CL model (p < 0.05 and p < 0.01). ALMA does not incorporate dropout mechanisms, leading to the lack of Model CL experimental results.

measures lexical overlap, as well as the recently introduced COMET score (Rei et al., 2022) based on the *wmt22-comet-da* model, which shows higher correlation with human judgments.

4.2 Consistency Learning Methods

Data Consistency Learning (Data CL) Data CL mainly forms different consistency sources at the data level. In the case of IT and TMT of WMT22 (ALMA), we induce perturbations by randomly substituting tokens in the input, thereby introducing elements of inconsistency. For TMT of IWSLT14 and MMT, we employ CipherDAug (Kambhatla et al., 2022) to perform character-level shifting on the source sentence. For ST, we randomly masked some speech frames with a certain probability (Chen et al., 2021a).

Model Consistency Learning (Model CL) The consistency sources in Model CL methods predominantly originate from variations in model structures or parameters. Across all the MT tasks, we utilize the widely adopted R-Drop method (Liang et al., 2021), which generates model variants through the application of random dropout masks. It is noted that ALMA and IT do not yet see advantages from Model CL, as the training and fine-tuning processes for LLMs do not incorporate dropout mechanisms.

Please refer to Appendix A.2, A.3 for more details of the datasets, method training, and inference.

4.3 Main Result

Performance Boost on IT As presented in Table 1, CCL significantly enhances the IT performance of LLMs. Unlike Q2Q (Li et al., 2023a),

Instruction Mining (Cao et al., 2023), and Alpa-Gasus (Chen et al., 2023), which depend on external data or models for curating high-quality data, CCL autonomously identifies appropriate samples for learning, leveraging the model's self abilities. While CL methods yield modest enhancements in LLM abilities (~ 0.5 points), our CCL method refines the learning strategy of CL, resulting in a substantial performance boost (~ 2.0 points).

Performance Boost on MT As depicted in Table 2, CCL yields considerable improvements across both unimodal and multimodal MT tasks. In TMT, the incorporation of CCL leads to advancements in both metrics. It also effectively applies to models of differing sizes, from the smaller scale Transformer to the larger language model ALMA. For MMT, CCL sets better results on the Multi30K dataset compared with the leading VL-T5 method. In ST, CCL surpasses the powerful model (Zhang et al., 2022) by achieving 0.9 COMET improvements. Most notably, CCL usually achieves marked improvements in the reliable metric COMET, indicating that it addresses more semantically complex difficulties, which tend to pose greater challenges.

5 Analysis

In this section, we aim to answer the following research questions: 1) Can CCL improve the efficiency and universality of CL? 2) Is CL loss a better difficulty measurer for CCL? 3) How does CCL enhance CSG model efficacy? We mainly evaluate four specific tasks, including TydiQA for IT on LLaMa-2-7B, IWSLT for TMT, Multi30K



Overall Method MMLU BBH GSM Tydiqa Foundation Model CodeX AE AVG $\Delta (\uparrow)$ 34.3 Vanilla IT 46.2 39.2 15.0 43.3 27.8 33.7 LLaMA-2-7B 47.6 CCL 40.8 15.5 48.4 30.7 35.6 36.4 +2.1Vanilla IT 55.7 47.3 31.0 49.1 41.8 46.5 45.2 LLaMA-2-13B CCL 49.3 34.0 53.6 42.7 47.6 47.1+1.955.6 Vanilla IT 52.5 51.7 33.5 51.1 54.7 43.1 47.8 Mistral-7B CCL 55.1 53.4 34.5 53.6 53.3 44.4 49.1 +1.3

Figure 2: Evolution of validation scores across various training steps or epochs.

Table 3: Results of IT with various foundation models. Codex and AE mean HumanEval and AlpacaEval benchmark respectively.

34.5

39.0

43.1

42.0

60.2

63.1

48.2

49.7

49.7

51.0

-

+1.3

52.3

52.4

Method	IT	TMT	MMT	ST
wictiou	F1 / Steps	BLEU / Steps	BLEU / Steps	BLEU / Steps
Baseline	43.3/1.2K	34.6 / 100K	44.9 / 20K	29.9 / 50K
CL	43.7/1.4K	37.3 / 300K	45.6 / 30K	30.2 / 70K
w/ CCL	48.2/0.6K	37.8 / 140K	45.9 / 22K	30.6 / 55K

LLaMA-3-8B

Vanilla IT

CCL

59.6

59.6

Table 4: Overall speedup results. "Steps" means the step of each model reaching the performance of CL.

for MMT, and MUST-C for ST. Additionally, we employ Model CL for MT and Data CL for IT, acknowledging the absence of Model CL in IT. For performance metrics, we apply BLEU scores to assess translation quality and F1 scores for IT, except where noted otherwise.

5.1 CCL: An Effective and Universal Method

Learning Curves Figure 2 indicates that CCL exhibits a more pronounced ascent due to the integration of curriculum learning, which facilitates the model's ability to grasp simpler instances during the initial training stages. As the training progresses, CCL surpasses the CL, attributable to the model's enhanced ability to concentrate on more challenging instances. These observations demonstrate that CCL not only augments model performance but also optimizes training efficiency.

Expedited Convergence CCL demonstrates a superior convergence rate in comparison to consistency learning methods. Table 4 shows that CCL enables models to produce equivalent or superior

results with fewer updates across IT and MT tasks. CCL facilitates a speed-up factor of approximately 1.79 times, thereby underlining its beneficial impact on the process of model optimization. Additionally, we give a thorough time comparison for each step and total time in model training on CCL and traditional CL in Appendix A.4.

Various Foundation Models This part primarily discusses the extensive applicability of CCL on foundational LLMs of different sizes and capabilities. Specifically, we apply CCL to the Mistral-7B and LLaMA-3-8B LLMs and present our results on the open-instruct benchmark aligned with the aforementioned test configuration. As shown in Table 3, CCL consistently enhances the overall performance of different LLMs. This is particularly evident in the GSM and AlpacaEval benchmarks, as CCL is capable of effectively modulating the focus during various stages of the training process, without relying on additional resources.

5.2 CL Loss: A Universal Difficulty Measurer

Predefined Difficulty Measurer Conventional curriculum learning approaches often employ task-specific difficulty measures, but CCL allows for a task-independent evaluation of instance difficulty directly from the model. Based on the above analysis of Section 3.1 and the tasks statement of Section 4, we contrast three commonly used difficulty measures, namely sentence length for text input of



Figure 3: Model performance by difficulty bucket measures by the proposed CL-aware difficulty measurer. CCL mainly improves the performance of challenging instances. The benchmarks for IT, TMT, MMT, and ST are TydiQA, IWSLT, Multi30K, and MUST-C respectively. For performance metrics, we apply BLEU scores to assess translation quality and F1 scores for IT.

Method	IT	TMT	MMT	ST
Baseline	43.3	34.6	44.9	29.9
CL	43.7	37.3	45.6	30.2
CCL	48.4	37.8	45.9	30.6
w/ Sentence Length	46.2	37.5	45.7	30.3
w/ Number of Object	-	-	45.7	-
w/ Frames of Speech	-	-	-	30.3
w/ NLL Loss	45.5	37.6	45.8	30.4

Table 5: Ablation study of CCL. The inapplicable difficulty measure to a task is marked with "-".

IT, TMT, MMT, and ST, the number of targets for image input of MMT, and the number of frames for speech signals of ST. As shown in Table 5, different difficulty measures can improve the model performance consistently which can be attributed to the rationality and effectiveness of the overall CCL framework. However, CCL demonstrates superior performance compared to existing difficulty measures across diverse CSG tasks in different modalities. This is because the difficulty measurer of CCL is based on model self-feedback, which is more flexible and reasonable than human-predefined conscious evaluation, such as sample length.

Automatic Difficulty Measurer One intuitive, task-independent means of estimating instance difficulty is by employing the NLL loss. However, through our preliminary study on model proficiency estimation, we discovered that as model proficiency increases, NLL loss correspondingly decreases throughout the training process. This inverse relationship generates significant ambiguity and instability when differentiating instance difficulty. In contrast, CCL remains stable throughout the training process, thereby enabling consistent differentiation of instance difficulty. As substantiated by Table 5, CLL consistently outperforms the NLL loss method across all tasks. It is noteworthy that all aforementioned predefined and au-



Figure 4: Model performance across samples of varying difficulties on TMT. Notably, improvements in both easy and medium samples occur primarily at the develop and later stages. The advancements of difficult samples are predominantly observed in the final stage.

tomatic measures still utilize our proposed model proficiency estimator and model update strategy.

5.3 CCL Excels in Tackling Challenges

Performance by Difficulty Bucket We further analyze the performance improvement from CCL by dissecting results based on test instance difficulty, as shown in Figure 3. We use TydiQA and CodeX as a mixed test set for IT due to their suitability for calculating consistency loss. For translation tasks, we divide the test set into three difficulty levels using the CL loss and calculate the evaluation score. Results show that CCL and CL perform similarly in simpler instances, but CCL often outperforms in more challenging ones.¹ This is because CCL allows the model to navigate a better parameter space for optimization, improving performance where CL loss is high. A case study of translating challenging sentences is in Appendix A.5.

Performance on Different Training Stages CCL enhances the model's handling of challenging samples by focusing more on later training stages.

¹In MT tasks, it is observed that more challenging samples often outperform the simpler ones in terms of performance metrics. This phenomenon can be attributed to the complexity of difficult sentences (with larger CL loss and longer lengths) which leads to a higher number of n-gram matches and, consequently, an increase in BLEU scores (Liu et al., 2020).

Yet, it raises the question of whether it overlooks simpler early-acquired knowledge. To explore this, we divided the training into four stages and evaluated the model's performance on different difficulty tests in the IWSLT translation task. Figure 4 shows consistent improvement on easier samples throughout training, this is because complex samples contain simpler elements. Thus, dealing with complex samples doesn't hinder the model's efficiency with easier ones. This highlights CCL's ability to wisely distribute learning resources based on sample difficulty, optimizing overall performance.

6 Related Works

6.1 Consistency Learning

Consistency learning, a method aimed at harmonizing model predictions from various sample perspectives, has been incorporated in several research works. R-drop (Liang et al., 2021) enhances the model performance across 18 datasets by addressing the layer-wise inconsistency brought about by random dropout. CipherDAug (Kambhatla et al., 2022) employs multi-source training with multiple ROT-k ciphertexts generated from different k values for the plaintext as well as the original parallel data to improve neural machine translation. With the continuous development of NLP, researchers have begun to apply this technology to LLMs. ACE (Fang et al., 2024) deploys a bidirectional consistency loss to train the decoding process for a multimodal large language model, mitigating overfitting issues. NPoE (Graf et al., 2024) and G2ST (Chen et al., 2024) adopt Kullback-Leibler divergence to respectively constrain the dual predictions of the main MOE model for a data poisoning backdoor attacks task, and to minimize inconsistency between training and inference in enhancing the performance of models within the e-commerce sector. Lastly, Consistency Matters (Zhao et al., 2024) improves LLMs consistency by generating varied responses through multiple LLMs in industrial deployments. Although consistency learning can potentially augment model capability, achieving model convergence often demands extensive training time due to the need to regulate model consistency distribution and inherent multi-objective optimization challenges (Liang et al., 2021).

6.2 Curriculum Learning

Curriculum learning (Bengio et al., 2009) introduces a human-like learning process for models,

progressively transitioning from simpler to complex samples, contingent on the model's ability to learn. The challenge lies in defining the sample complexity and model capability. While sentence length (Platanios et al., 2019) has been utilized for the former, it is limited to text-related instances. Other research has used the norm value of the training corpus to characterize sample complexity and model capability (Liu et al., 2020). Techniques incorporating domain knowledge, LLMs' generative abilities, and an integrated curriculum scheduler have been effective in training a genetic programming agent player (Jorgensen et al., 2024). Stronger LLM teachers creating a structured curriculum for student LLMs have demonstrated notable improvements in language generation quality, evaluated by GPT-4 (Feng et al., 2023). The instructional tuning phase in ICCL (Liu et al., 2024) progressively adjusts prompt complexity, leading to better LLM's performances. Strategies such as dataset decomposition (to the union of buckets) and variable batch size utilization in conjunction with a curriculum have proven to be time-efficient in LLM training (Pouransari et al., 2024). (Erak et al., 2024) show the automation of curriculum design leveraging LLMs' generative capabilities, successfully improving the agent's performance and convergence rate while reducing manual effort. Despite advances, limitations like high manual intervention and the need for sturdy guiding models affect the scalability of these methods. How to optimally use CL in LLMs instructional tuning for dynamic sample recognition is still an unresolved issue. In this paper, we propose an innovative complementary approach that leverages the strengths of both consistency learning and curriculum learning to enhance the capabilities of CSG.

7 Conclusion

This paper introduces curriculum consistency learning (CCL), a novel approach designed to refine the optimization process of conditional sentence generation models with consistency learning. Our method proposes a novel mechanism for measuring difficulty and estimating model proficiency, grounded in the analysis of consistency learning loss observed during model training, which is userfriendly and operates autonomously. Extensive experiments confirm the effectiveness of CCL in improving model performance and tackling challenges from complex instances.

Limitations

While the proposed curriculum consistency learning (CCL) method can significantly optimize traditional CL methods, it has the following limitations: (1) Compared with traditional CL, CCL necessitates the inclusion of extra parameters (denoted as λ) to govern the curriculum's duration. (2) CCL's effectiveness relies on the joint evaluation of the training and validation sets, posing challenges for tasks without a dedicated validation set. (3) Presently, CCL is primarily applied to supervised learning tasks. Future efforts will aim to adapt CCL for unsupervised learning scenarios.

Ethics Statement

Our work follows the ACL Ethics Policy. Our findings are based on publicly available datasets for reproducibility purposes. LLMs can contain potential racial and gender bias. Therefore, if someone finds our work interesting and would like to use it in a specific environment, we strongly suggest the user check the potential bias before usage. In addition, it is hard to control the generation of LLMs. We should be aware of the potential problems caused by hallucinations.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62206076), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011491), Shenzhen Science and Technology Program ZDSYS20230626091203008, (Grant Nos. KJZD20231023094700001,

RCBS20221008093121053), and Shenzhen College Stability Support Plan (Grant Nos. GXWD20220811173340003, GXWD20220817123150002). We would like to thank the anonymous reviewers and meta-reviewer for their insightful suggestions.

References

- Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *ArXiv preprint*, abs/2305.10403.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In

Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009, volume 382 of ACM International Conference Proceeding Series, pages 41–48. ACM.

- Yihan Cao, Yanbin Kang, and Lichao Sun. 2023. Instruction mining: High-quality instruction data selection for large language models. *ArXiv preprint*, abs/2307.06290.
- Jiaao Chen, Dinghan Shen, Weizhu Chen, and Diyi Yang. 2021a. HiddenCut: Simple data augmentation for natural language understanding with better generalizability. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4380–4390, Online. Association for Computational Linguistics.
- Kaidi Chen, Ben Chen, Dehong Gao, Huangyu Dai, Wen Jiang, Wei Ning, Shanqing Yu, Libin Yang, and Xiaoyan Cai. 2024. General2specialized llms translation for e-commerce. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 670–673.
- Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *ArXiv preprint*, abs/2307.08701.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021b. Evaluating large language models trained on code. *ArXiv preprint*, abs/2107.03374.
- Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, pages 1931–1942. PMLR.
- Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168.
- Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In Proceedings of the 2019 Conference of the North American

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

- Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback.
- Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70– 74, Berlin, Germany. Association for Computational Linguistics.
- Omar Erak, Omar Alhussein, Shimaa Naser, Nouf Alabbasi, De Mi, and Sami Muhaidat. 2024. Large language model-driven curriculum design for mobile networks.
- Run-Ze Fan, Xuefeng Li, Haoyang Zou, Junlong Li, Shwai He, Ethan Chern, Jiewen Hu, and Pengfei Liu. 2024. Reformatted alignment. In *Findings of the Association for Computational Linguistics: EMNLP* 2024.
- Minghui Fang, Shengpeng Ji, Jialong Zuo, Hai Huang, Yan Xia, Jieming Zhu, Xize Cheng, Xiaoda Yang, Wenrui Liu, Gang Wang, et al. 2024. Ace: A generative cross-modal retrieval framework with coarse-to-fine semantic modeling. *arXiv preprint arXiv:2406.17507*.
- Qingkai Fang and Yang Feng. 2023. Understanding and bridging the modality gap for speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15864–15881, Toronto, Canada. Association for Computational Linguistics.
- Tao Feng, Zifeng Wang, and Jimeng Sun. 2023. Citing: Large language models create curriculum for instruction tuning. arXiv preprint arXiv:2310.02527.
- Victoria Graf, Qin Liu, and Muhao Chen. 2024. Two heads are better than one: Nested PoE for robust defense against multi-backdoors. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 706–718, Mexico City, Mexico. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net.

- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022.* OpenReview.net.
- Steven Jorgensen, Giorgia Nadizar, Gloria Pietropolli, Luca Manzoni, Eric Medvet, Una-May O'Reilly, and Erik Hemberg. 2024. Large language model-based test case generation for gp agents. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 914–923.
- Nishant Kambhatla, Logan Born, and Anoop Sarkar. 2022. CipherDAug: Ciphertext based data augmentation for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 201–218, Dublin, Ireland. Association for Computational Linguistics.
- Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, pages 379–386, Varna, Bulgaria. INCOMA Ltd.
- Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.
- Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023a. From quantity to quality: Boosting Ilm performance with self-guided data selection for instruction tuning. *ArXiv preprint*, abs/2308.12032.
- Yi Li, Rameswar Panda, Yoon Kim, Chun-Fu Richard Chen, Rogério Feris, David D. Cox, and Nuno Vasconcelos. 2022a. VALHALLA: visual hallucination for machine translation. In *IEEE/CVF Conference* on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pages 5206–5216. IEEE.
- Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F. Wong, Yang Gao, Heyan Huang, and Min Zhang. 2023b. TemplateGEC: Improving grammatical error correction with detection template. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6878–6892, Toronto, Canada. Association for Computational Linguistics.
- Zhaocong Li, Xuebo Liu, Derek F. Wong, Lidia S. Chao, and Min Zhang. 2022b. ConsistTL: Modeling consistency in transfer learning for low-resource neural machine translation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8383–8394, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

- Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 10890– 10905.
- Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theory*, 37(1):145–151.
- Shudong Liu, Xuebo Liu, Derek F. Wong, Zhaocong Li, Wenxiang Jiao, Lidia S. Chao, and Min Zhang. 2023. kNN-TL: k-nearest-neighbor transfer learning for low-resource neural machine translation. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1878–1891, Toronto, Canada. Association for Computational Linguistics.
- Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the* 58th Annual Meeting of the Association for Computational Linguistics, pages 427–436, Online. Association for Computational Linguistics.
- Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, and Wei Lu. 2024. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiv preprint arXiv:2402.10738*.
- Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. ArXiv preprint, abs/2304.03277.
- Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.
- Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Hadi Pouransari, Chun-Liang Li, Jen-Hao Rick Chang, Pavan Kumar Anasosalu Vasu, Cem Koc, Vaishaal Shankar, and Oncel Tuzel. 2024. Dataset decomposition: Faster Ilm training with variable sequence length curriculum. arXiv preprint arXiv:2405.13226.

- Jun Rao, Liu Xuebo, Lian Lian, Shengjun Cheng, Yunjie Liao, and Min Zhang. 2024. Commonit: Commonality-aware instruction tuning for large language models via data partitions. In *EMNLP*.
- Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Petru Soviany, Claudiu Ardei, Radu Tudor Ionescu, and Marius Leordeanu. 2020. Image difficulty curriculum for generative adversarial networks (cugan). In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3463–3472.
- Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526– 1565.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *ArXiv preprint*, abs/2210.09261.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.
- Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576.
- Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023. How far can camels go? exploring the state of instruction tuning on open resources. ArXiv preprint, abs/2306.04751.
- Shufang Xie, Ang Lv, Yingce Xia, Lijun Wu, Tao Qin, Tie-Yan Liu, and Rui Yan. 2022. Target-side input augmentation for sequence to sequence generation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-*29, 2022. OpenReview.net.

- Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations*.
- Yongjing Yin, Jiali Zeng, Yafu Li, Fandong Meng, Jie Zhou, and Yue Zhang. 2023. Consistency regularization training for compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1294–1308, Toronto, Canada. Association for Computational Linguistics.
- Ziqiang Zhang, Long Zhou, Junyi Ao, Shujie Liu, Lirong Dai, Jinyu Li, and Furu Wei. 2022. SpeechUT: Bridging speech and text with hiddenunit for encoder-decoder based speech-text pretraining. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1663–1676, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Fufangchen Zhao, Guoqiang Jin, Jiaheng Huang, Rui Zhao, and Fei Tan. 2024. Consistency matters: Explore llms consistency from a black-box perspective. *arXiv preprint arXiv:2402.17411*.



Figure 5: CL loss, derived automatically from the model, aligns closely with unique feature metrics across various domains that need to be designed by human experts. These results fully demonstrate the rationality of CL loss as a reasonable difficulty measurer of curriculum learning.

Configuration	TMT		MMT	ST	IT		
Comgulation	IWSLT	WMT22	Mutil30K	MUST-C	Alpaca	Alpaca	
Architecture	Transformer-base	ALMA-7B	T5-base	SpeehUT	LLaMA-2-7B	LLaMA-2-13B	
Encoder layer	6	-	12	12	-	-	
Decoder layer	6	32	12	6	32	40	
Model dimension	512	4096	512	768	4096	5120	
Attention head	4	32	12	8	32	40	
Steps	300K	4200	30K	50K	1400	1400	
Max tokens / Batch size	4,096	8	30	800K	128	128	
Learning rate	5e-5	2e-5	5e-5	3e-5	2e-5	2e-5	
Optimizer	Adam	AdamW	AdamW	Adam	AdamW	AdamW	
Dropout	0.3	-	0.3	0.3	-	-	
Temperature	-	0.1	-	-	0.1	0.1	
Beam size	5	4	5	10	4	4	
Length penalty	1.0	1.0	1.0	1.0	1.0	1.0	
α	5	0.0001	0.001	0.9	0.00012	0.00015	

Table 6: Hyper-parameters of different models. CCL directly inherit all the hyper-parameters from the baselines and CL methods, without any additional tuning.

A Appendix

A.1 Task-specific Difficulty Measurer

Our proposed CL-aware difficulty metric presents distinctive and compelling advantages. To further illustrate these, we compared the CL loss with other prevalent task-specific difficulty measures across various modalities, using four distinct tasks for detailed task descriptions). We calculate the parse tree depth and sentence length for all the text sentence input. Particularly, taking into account the specificity of multimodal tasks, we also estimate the number of objects for the image data input of MMT and the frames of speech for the speech signals of ST. Then we randomly selected 1,000 instances from the training set and arranged them in ascending order based on the average CL loss throughout the entire training process. Subsequently, we divided them evenly into five groups, using the group with the highest value as the difficulty reference and setting the difficulty coefficient to 1. As shown in Figure 5, we found that the CL loss exhibits strong alignment with distinct feature

metrics of each modality, which demonstrates that our difficulty indicator is independent of the input modality and showcases its widespread applicability as a difficulty measurer for curriculum learning in CL-enhanced models.

A.2 Datasets and Settings

Instruction Tuning (IT) In CCL, LLMs are developed on the Alpaca GPT-4 (Peng et al., 2023). Specifically, we partition the Alpaca dataset into two distinct sets: a validation set comprising 10% of the total data, and a training set containing the remaining 90%, approximately 5.2K and 46.8K instances, respectively. We evaluate our IT models on the popular code repository: Open-Instruction (Wang et al., 2023), which includes an assessment of Factual knowledge, Reasoning, Multilinguality, Coding, and other abilities. For the MMLU component, we adapt the original MMLU framework's approach, presenting results based on five few-shot examples. In the case of the BBH and the GSM evaluations, we implement different few-shot in-context setups: three for BBH and

			Alpaca IT Task			
U	al input ented input		a short story in third person narration about a protagonist who has to make an important career decision. a red story in third person book about a protagonist who has to write an important career decision.			
Target	Farget Sophie sat at her desk, staring blankly at the computer screen. Her mind was racing as she weighed the o					
			Table 7: Different views on IT.			
			IWSLT14 German-English TMT Task			
	Original input Augmented input		Jeder Vorsitzende bzw. jede Vorsitzende hat ja so viele Stimmen, wie die Fraktion Mitglieder hat. Kfefs wpstjuafoef cax. kfef wpstjuafoef ibu kb tp wjfmf tujnnfo, xjf ejf gsblujpo njuhmjfefs ibu.			
	Target		As you know, each chairman has the same number of votes as his Group has Members.			
	WMT22 English-Chinses TMT Task					
	Original inp Augmented		We still have a lot of work to do in this area as recent events have proved. We still not a lot of work to do in this case as recent events have proved.			
	Target		最近的事件表明,大家在方面仍有多工作要做。			

Table 8: Different views on TMT.

eight for GSM, both employing Chain-of-Thought (CoT) reasoning. For the TydiQA evaluation, we follow the setup described in the PaLM 2 technical report (Anil et al., 2023) to evaluate models' performance in answering multilingual questions when there is context given. Regarding the CodeX evaluation, we leverage the HumanEval (Chen et al., 2021b) dataset to gauge the coding abilities of the models. We report pass@10 results which were obtained by sampling with a temperature of 0.8.

Textual Machine Translation (TMT) We conduct experiments on IWSLT14 German \rightarrow English and WMT14 English \rightarrow German machine translation tasks. The IWSLT14 dataset² consists of approximately 170,000 sentence pairs dedicated to training purposes. This is supplemented with additional 7,000 sentence pairs each for the validation and testing stages of our experimental procedure. The WMT14 dataset³ is much larger, incorporating an impressive count of 4.5 million sentence pairs for training. The validation and test sets are newstest2013 and nestest2014, respectively.

Multimodal Machine Translation (MMT) We assess the performance of multimodal machine translation on the Multi30K dataset, which consists of training, validation, and test subsets comprising 29,000, 1,014, and 1,000 instances, respectively. The Multi30K dataset is specifically curated for

tasks pertaining to machine translation and image captioning. It is essentially an extension of the Flickr30K dataset and includes image-description pairings across five unique languages - English, French, German, Czech, and Russian, thereby establishing its multilingual nature.

Speech-to-Text Translation (ST) The MUST-C dataset, a renowned and broadly employed resource in speech-to-text (ST) tasks boasts an extensive and heterogeneous corpus. It offers a generous allocation of approximately 400 hours of speech per language, providing a rich, comprehensive repository of multilingual data dedicated to ST tasks. This collection emphasizes direct translations from English into eight distinct languages. Further augmenting its utility, the dataset consists of training, validation, and test subsets comprising 220K, 1.4K, and 2.6K instances, which are specifically tailored for English-to-German translation exercises.

A.3 Data Consistency Learning

Instruction Tuning (IT) We employ HiddenCut (Chen et al., 2021a) technique to randomly replace the token with 1% probability for the original input, generating multiple views of the same sample, as shown in Table 7.

Textual Machine Translation (TMT) For the TMT of IWSLT14, we employ Cipher-DAug (Kambhatla et al., 2022) to transform the source side of the dataset, creating two different views for each sample. As shown in Table 8, each character on the source side shifts from one position to the right while maintaining the target side in

²https://github.com/facebookresearch/fairseq/ blob/main/examples/translation/prepare-iwslt14. sh

³https://github.com/facebookresearch/ fairseq/blob/main/examples/translation/ prepare-wmt14en2de.sh





Multi30K English-German MMT Task					
Original input Augmented input	A man is smiling at a stuffed lion. B nbo jt tnjmioh bu b tuvggfe mjpo.				
Target	Ein Mann lächelt einen ausgestopften Löwen an.				

Original input

Augmented input

Figure 6: Different views on MMT.



Figure 7: Different views on ST.

its original form. For the TMT of ALMA, we follow HiddentCut (Chen et al., 2021a) to randomly replace the token with 1% probability for the original input, generating multiple views of the same sample.

Multimodal Machine Translation (MMT) We employ the CipherDAug technique to transform the source data of text and apply the horizontal flip on the source data of images, generating different views of the same sample, as shown in Figure 6.

Speech-to-Text Translation (ST) Inspired by the idea of HiddenCut (Chen et al., 2021a), we mask audio frames with 1% probability during the training process, generating multiple views of the same sample, as shown in Figure 7.

A.4 Training Speed Comparison

To highlight CCL's efficiency, we compared the model's processing time across different tasks. Our study reveals that both CCL and traditional CL require similar durations per step because our proposed CCL method only alters the loss weights of different samples while leaving all other elements intact. This guarantees that the time spent on each training step for both CCL and traditional CL remains relatively equal. To substantiate our results, we have included comprehensive data regarding training speeds and total time used for various tasks in Tables 10 to 13. Lastly, the IT task evaluation was conducted with 4 A800 GPUs, while we used 4 3090 GPUs for the remaining tasks.

A.5 Case Study

Table 9 presents a translation example of challenging sentences whose consistency loss is high. We observe that CL merely translates the short portions at the end of the sentences without ensuring the correctness of the sentence tense. In contrast, CCL successfully translates the entire sentence, ensuring the correctness of the sentence's semantics and tense. This can be attributed to CCL increasing the loss weight of difficult instances in the later stages of training, further demonstrating that CCL performs better in handling difficult instances.

Source	Ich habe mich sofort seinem Bruder und seinem Vater verpflichtet. Ich sagte: "Also, hier ist der Deal: Tony wird sprechen.
Reference	Wir werden ihm eine Maschine besorgen und einen Weg finden, damit er wieder seine Kunst machen kann." I committed to his brother and his father right then and there. I said, "I am like, here's the deal". Tony's going to speak. We are going to get him a machine, and we are going to figure out a way for him to do his art again.
CL	I committed myself to the place and to his brother and father, saying, "Well, my suggestion is this: Tony will talk.
	We get him a device, and we find a way to do his art again."
CCL	I am committed to the place, and I am committed to his brother and his father. Well, my proposal is the following: "Tony's going to talk. We're going to get him a machine, and we're going to find a way to make his art again."

Table 9: An example of TMT that is regarded as a difficult instance.

Method	Training Speed (Seconds/Steps)	Steps(K)	Time(H)
Vanilla	16.5	1.2	5.5
CL	21.6	1.4	8.4
CCL	21.9	0.6	3.7

Table 10: Comparison of training speed of different methods in Instruction Tuning (IT).

Method	Training Speed (Seconds/Steps)	Steps(K)	Time(H)
Vanilla	0.14	100	3.8
CL	0.15	300	12.2
CCL	0.15	140	5.8

Table 11: Comparison of training speed of differentmethods in Textual Machine Translation (TMT).

Method	Training Speed (Seconds/Steps)	Steps(K)	Time(H)
Vanilla	0.85	20	4.7
CL	1.03	30	8.6
CCL	1.04	22	6.4

Table 12: Comparison of training speed of different methods in Multimodal Machine Translation (MMT).

Method	Training Speed (Seconds/Steps)	Steps(K)	Time(H)
Vanilla	0.74	50	10.3
CL	0.95	70	18.4
CCL	0.95	55	14.5

Table 13: Comparison of training speed of differentmethods in Speech-to-Text Translation (ST).