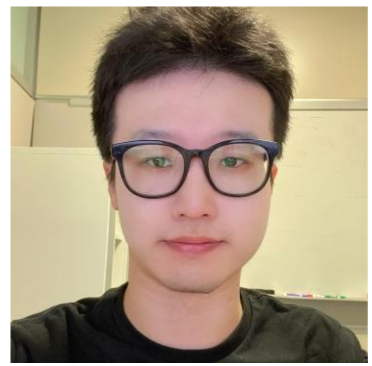
DistiLLM: Towards Streamlined Distillation for Large Language Models

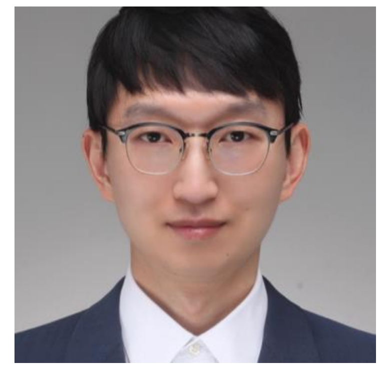
ICML 2024

Jongwoo Ko
KAIST AI

Sungnyun Kim
KAIST AI



Tianyi Chen
Microsoft



Se-Young Yun
KAIST AI

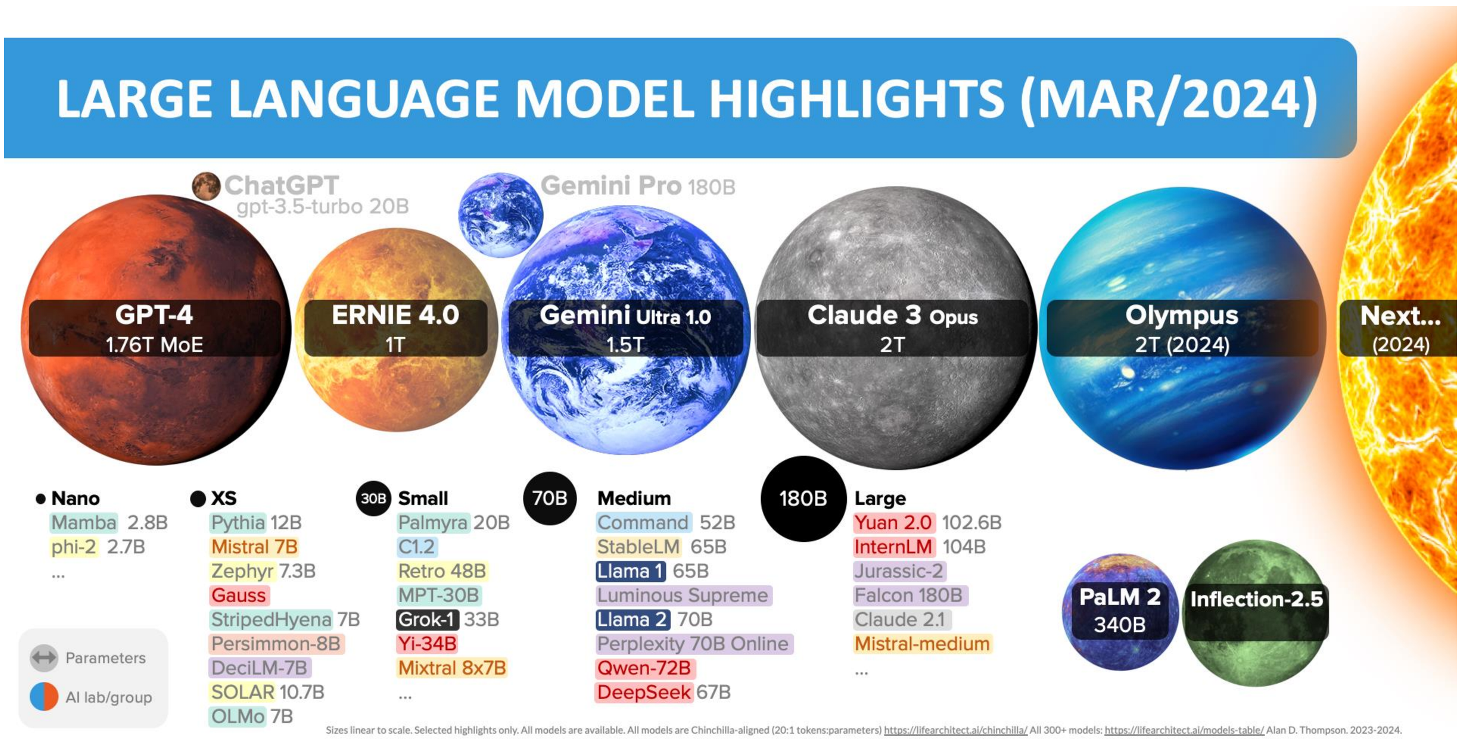
KAIST AI
Graduate School of AI

Microsoft

ICML
International Conference
On Machine Learning

Large Language Models

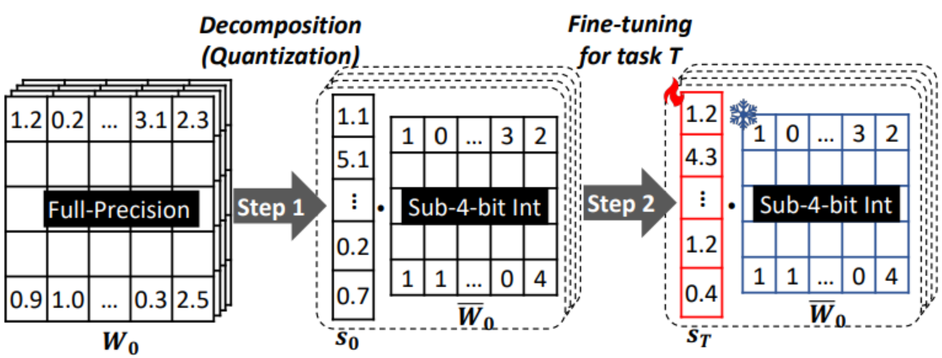
- **Large Language Models**
- LLMs have **significantly improved** the **quality of generation**
- Attributed to the **increased scale** of **training data** and **model parameters**.
- **Higher inference costs** or **large memory footprints**



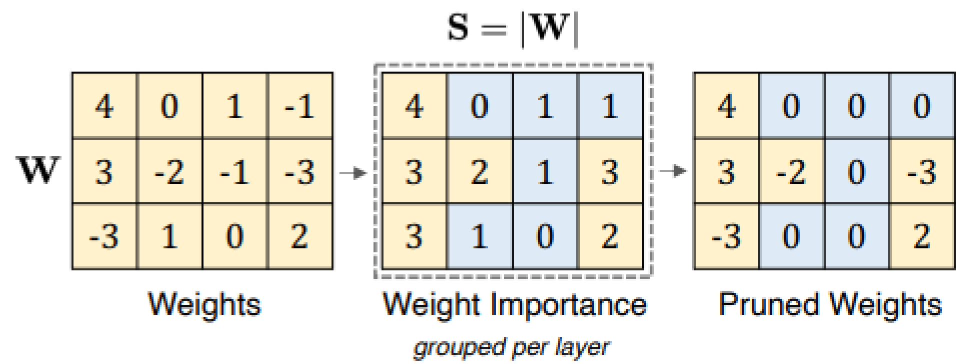
[LifeArchitect.ai/models]

LLM Compressions

- Necessity of reducing the demands on computational resources becomes important
- **Quantization**: Making weights and activation into **low-bit** integers (i.e., 3-bit, 4-bit)
- **Network Pruning**: Remove **redundant** units (i.e., neuron, head, block) of network
- **Knowledge Distillation**: building **small student** models that can mimic larger model
- **Inference Acceleration**, **Mixture-Of-Expert**, …



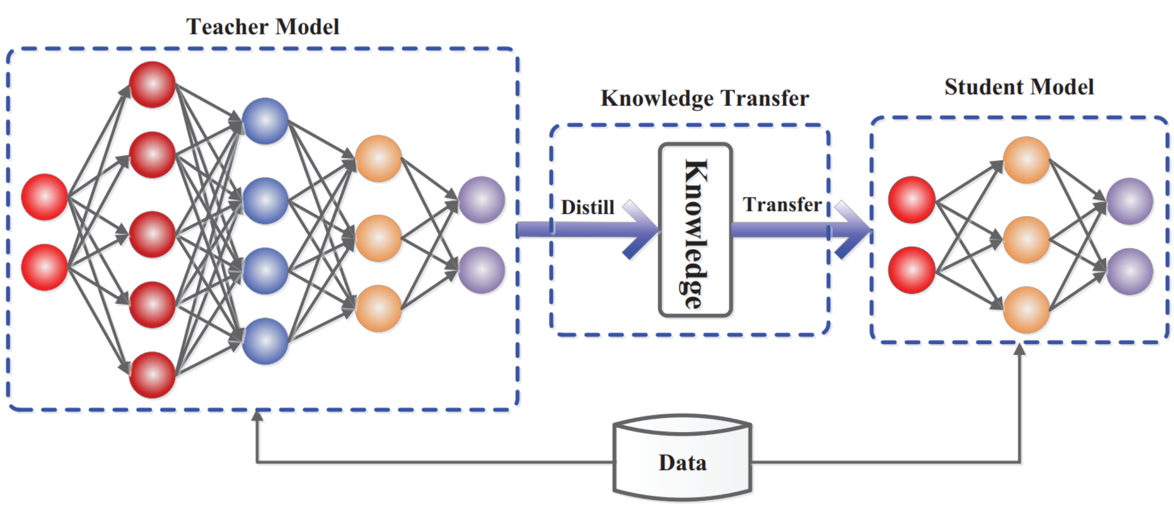
Quantization

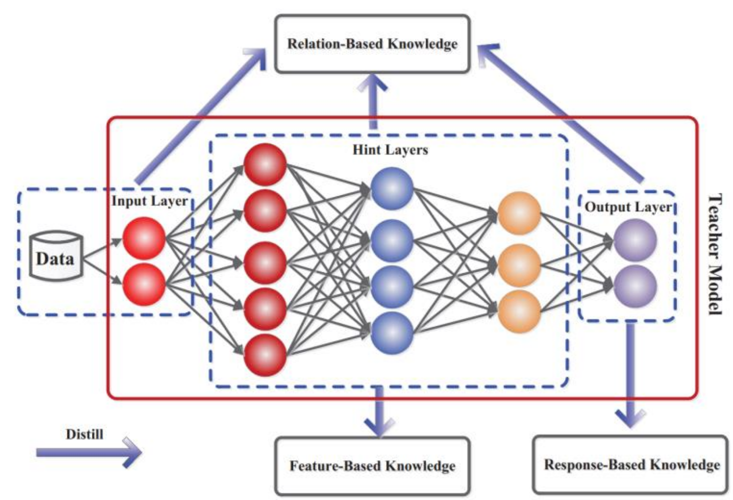


Network Pruning

[Kim et al. NeurIPS (2023); Sun et al., ICLR (2024)]

- **Knowledge Distillation**
- Making smaller **student models** that **mimic** the response of larger teacher models.
- **Saving computational resources** with **minimal performance reduction**
- A vanilla KD uses the **logits** of a large deep model as the teacher knowledge.
- The **activations**, neurons or **features** of intermediate layers also can be used as the knowledge to guide the learning of the student model.

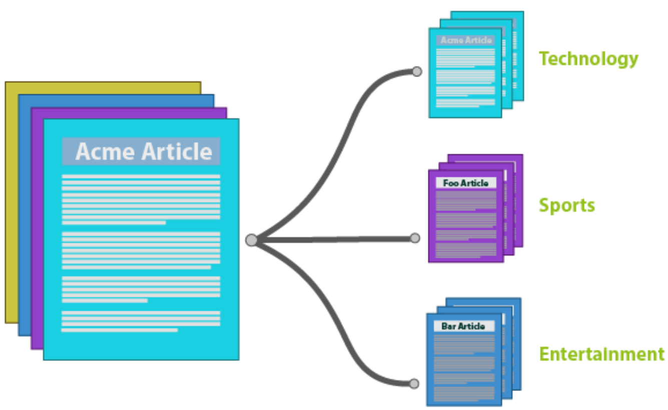




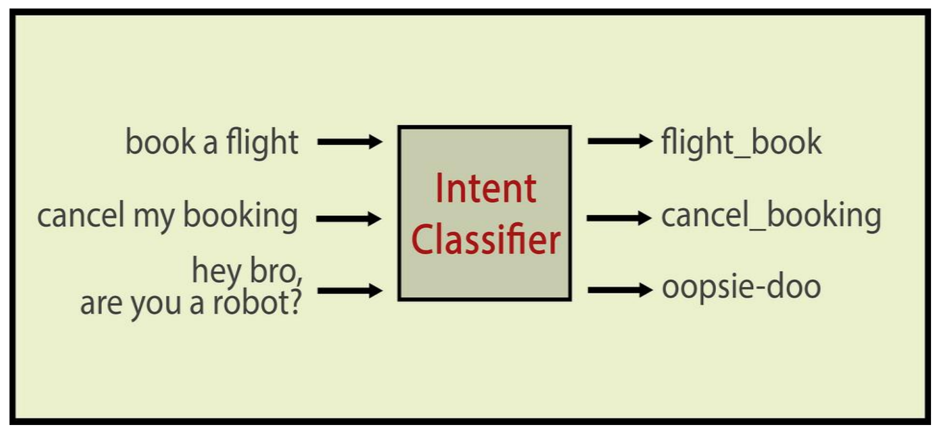
[Gou et al., IJCV (2021)]

✓ Discriminative LMs (BERT, RoBERTa, ELECTRA)

- KD approaches in NLP, are mostly studied for small (< 1B parameters) discriminative LMs.
- Due to small model size, such models can utilize better signals from output distribution and hidden states of teacher models.
- In LLMs, this is not applicable in common.

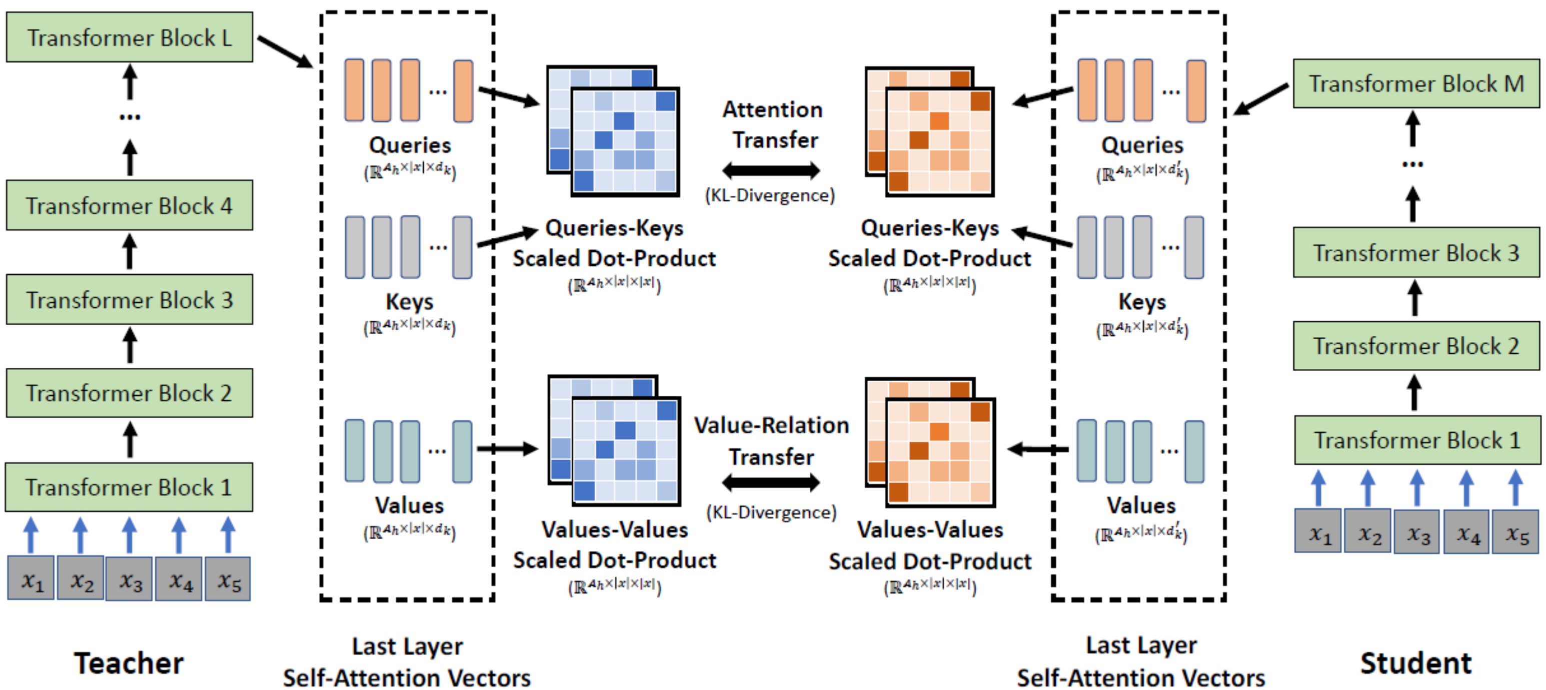


Document Classification



Intent Classification

- **Hidden Representation Distillation**
- TinyBERT (Huawei), MobileBERT (Google), MiniLM (Microsoft Research)
- Using the **hidden representations** or **attention mapping**
- Showing effectiveness for BERT (both pre-training & fine-tuning)



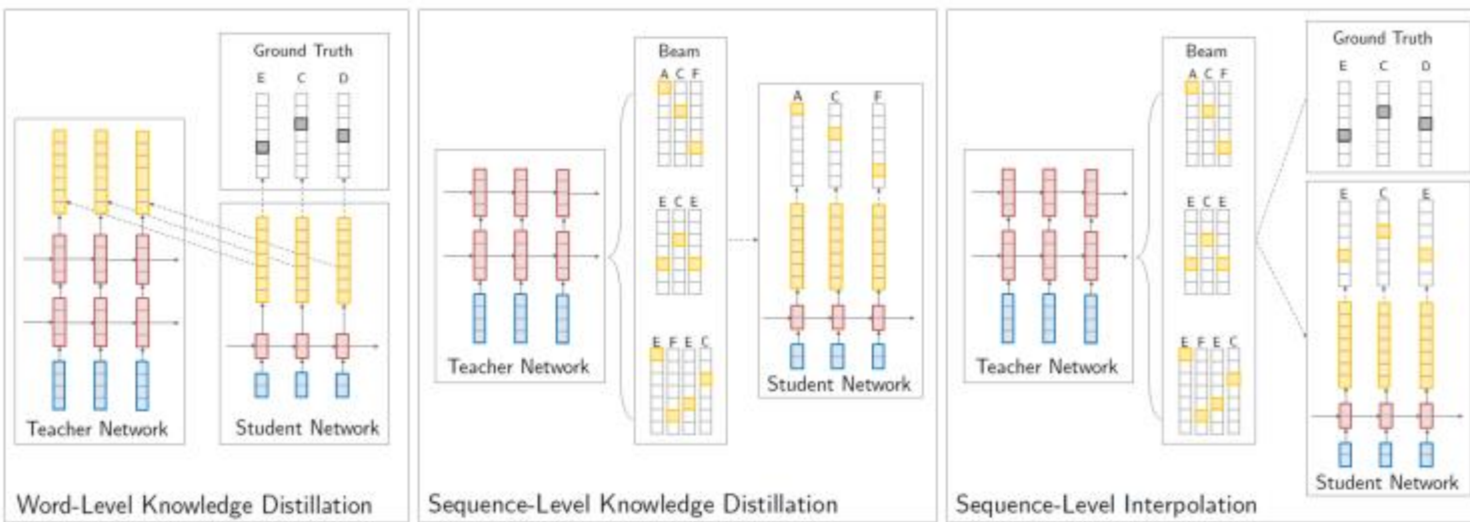
[Wang et al., NeurIPS 20']

Generative LMs (GPT-4, Claude-3, Gemini)

- **Larger output space** than classification task
  - For **text (or image) classification**, KL divergence works well because the output space is quite **small**.
  - At most, **1K classes** for classification (ImageNet) vs. **vocab size of 30K ~ 250K for LLMs**
- **Training-Inference mismatch**
  - Generative LMs train in **teacher-forcing** manners, however, inferences in **auto-regressive** manners.
  - Also known as **exposure bias**

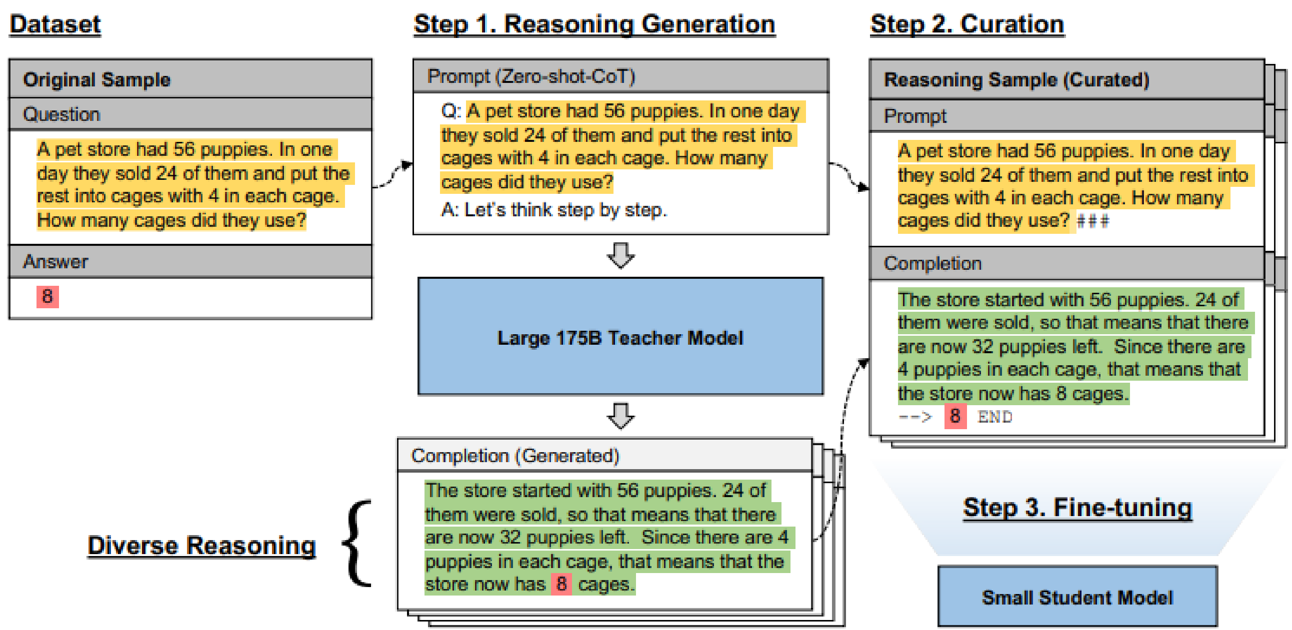
SeqKD (Sequence-Level Knowledge Distillation)

- **Sequence-Level Knowledge Distillation (EMNLP 16')**
- **Train** the student network w/ cross-entropy on the **teacher model generation**.
- (1) Train teacher model (2) Run beam search over the training set (3) train the student network w/ CE on this new dataset.
- 10 times faster than SOTA teacher with little loss in performance.



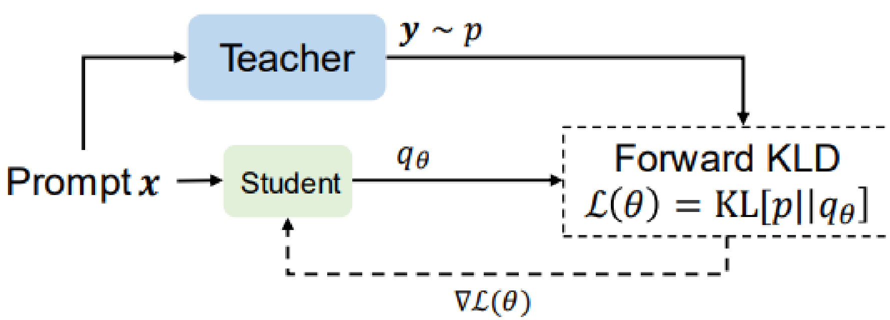
[Kim and Rush, EMNLP 16']

- **Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes (ACL 23')**
- **Large Language Models Are Reasoning Teachers (ACL 23')**
- SeqKD recently get popularity in LLM era, especially for **closed-source LLMs**. (**Black-box KD**)
- Small LMs can get **reasoning abilities** which is known as **emergent ability of LLMs**.

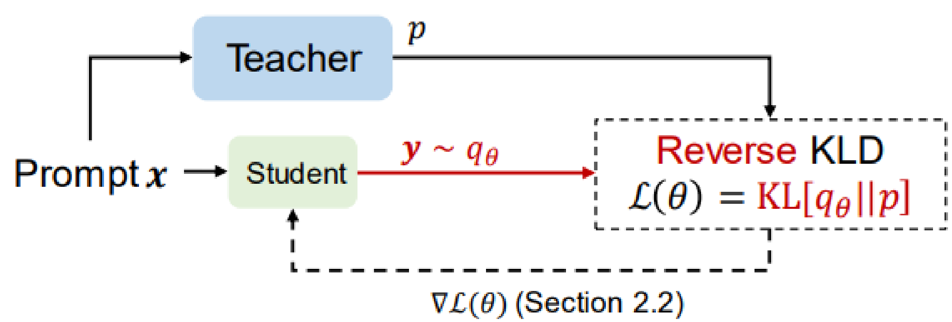


[Ho et al., ACL 23']

- **MiniLLM: Knowledge Distillation of Large Language Models (ICLR 24')**
- Sequence-level KD into **reinforcement learning framework**.
- Using **reverse KL** divergence: $\theta = \arg\min \mathcal{L}(\theta) = \arg\min_{\theta} \mathrm{KL}(q_\theta || p)$
- **Policy gradient Theorem**: $\nabla \mathcal{L}(\theta) = \sum_{t=1}^{T} (R_t - 1) \nabla \log q_\theta(y_t | \boldsymbol{y}_{<t}, \boldsymbol{x})$
- + Additional Technique to **instability problems** of policy gradient
- **Single-step decomposition** / **Techer-mixed sampling** / **Length normalization**



Sequence-Level KD



MiniLLM (Ours)

[Gu et al., ICLR 24']

KAIST AI Graduate School of AI ICML International Conference On Machine Learning

- **On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes (ICLR 24')**
- Using **student-generated outputs (SGO)** for addressing **train-inference mismatch**.
- Motivated by on-policy imitation learning, popular in robotics and deep RL.
- Student receives token-specific feedback from the teacher's logits on **erroneous tokens**.

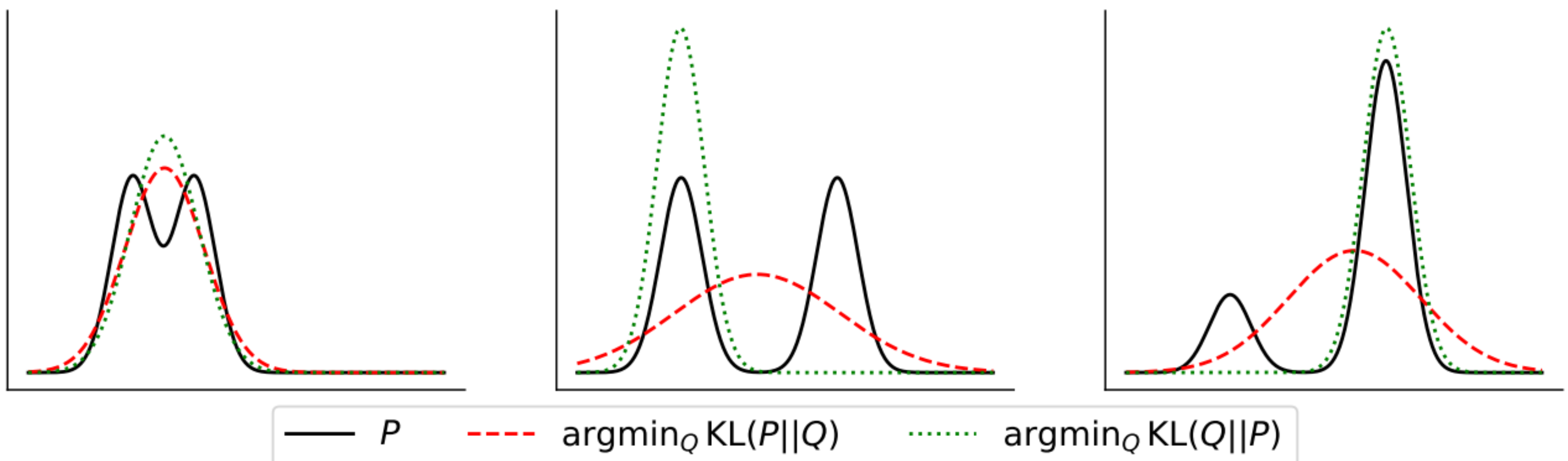
Algorithm 1 Generalized Knowledge Distillation (GKD)

1: **Given**: Teacher model p_{T}, Student Model p_{S}^{θ}, Dataset (X, Y) containing (input, output) pairs
2: **Hyperparameters**: Student data fraction $\lambda \in [0, 1]$, Divergence $\mathcal{D}$, Learning rate η
3: **for** each step $k = 1, \ldots, K$ **do**
4: Generate a random value $u \sim Uniform(0, 1)$
5: **if** $u \leq \lambda$ **then**
6: Sample inputs x from X and generate outputs $y \sim p_{\mathrm{S}}^{\theta}(\cdot|x)$ to obtain $B = \{(x_b, y_b)\}_{b=1}^{B}$
7: **else**
8: Sample batch of inputs and outputs from (X, Y) to obtain $B = \{(x_b, y_b)\}_{b=1}^{B}$.
9: **end if**
10: Update θ to minimize L_{GKD}: $\theta \leftarrow \theta - \eta \frac{1}{B} \sum_{(x,y) \in B} \nabla_{\theta} \mathcal{D}(p_{\mathrm{T}} \| p_{\mathrm{S}}^{\theta})(y|x)$
11: **end for**

[Agarwal et al., ICLR 24']

Problem of Recent KD for LLMs (1)

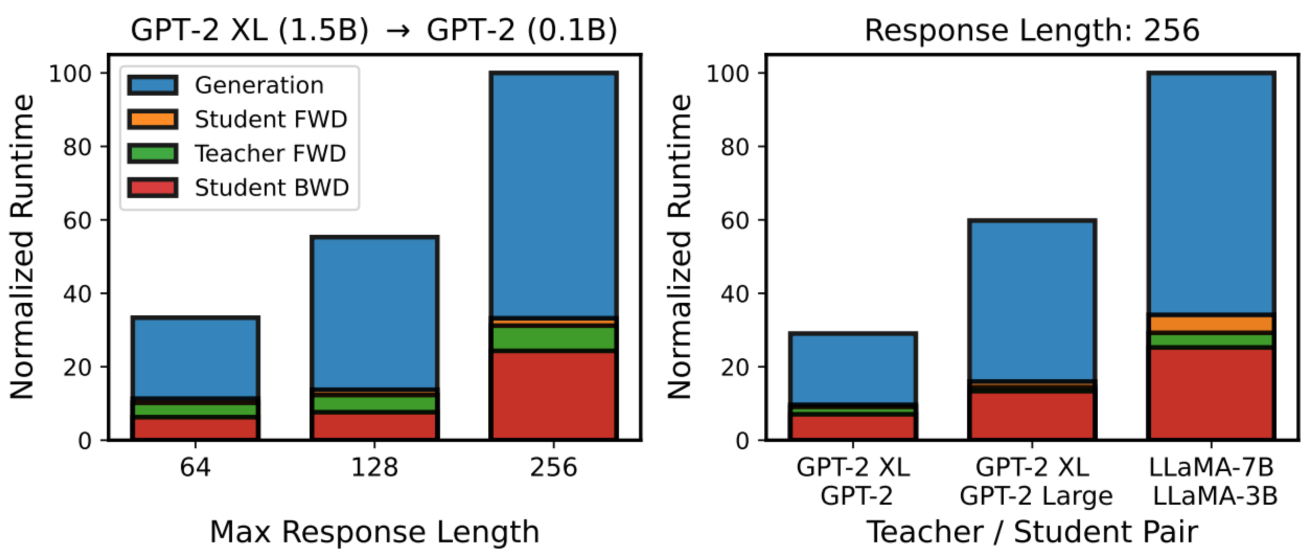
- **Lack of in-depth analysis for objective functions**
- MiniLLM used policy gradient to minimize reverse KLD.
- GKD and f-distill evaluated various objective functions: (reverse) KLD, JSD, TVD
- Results indicated the **optimal divergence** seems to be **task-dependent**.
- **Requiring additional efforts** to inconveniently select a proper loss function.



[Agarwal et al., ICLR 24']

Problem of Recent KD for LLMs (2)

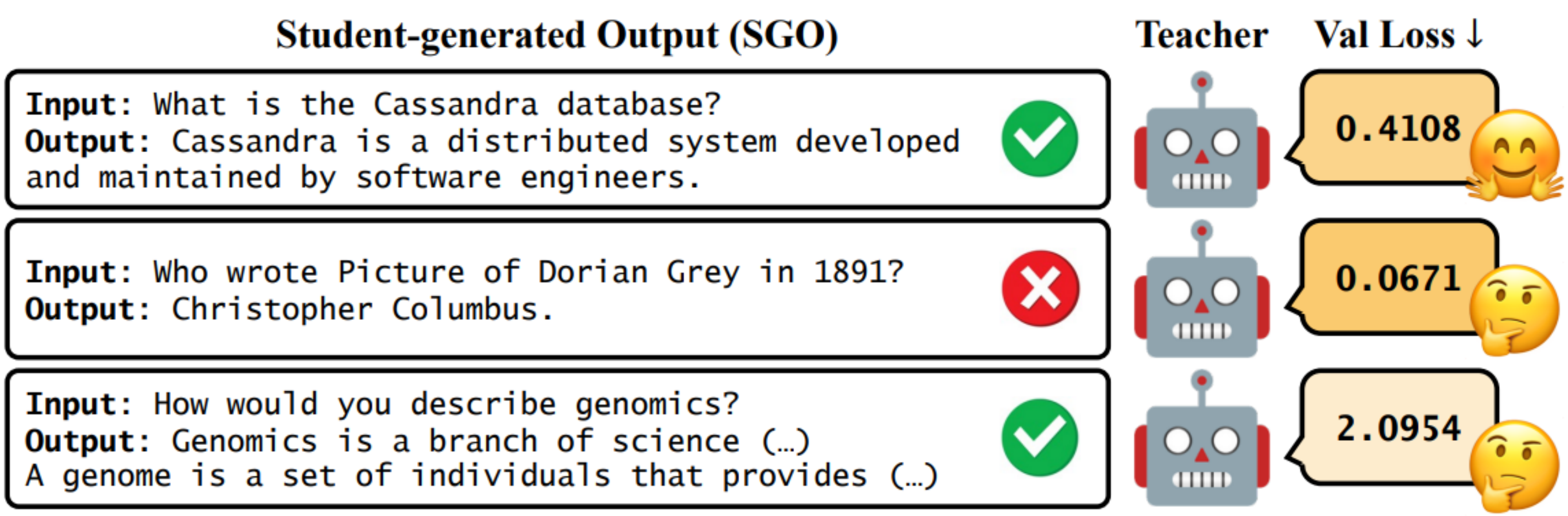
- **Heavy computation of SGO**
- On-policy distillation has been shown effectiveness in recent studies.
- However, **generating SGOs** for every iteration is **computationally inefficient**.
- SGO generation accounts for a consideration portion of the total training time, **reaching up to 80%**.



Problem of Recent KD for LLMs (3)

Negative Effect of SGOs

- The **inaccurate** or **unfamiliar SGOs** to teacher model potentially lead to **misguidance**.
- MiniLLM suggested to mix the distribution of teacher and student to alleviate this.
- However, this notably increases training computation because teacher model is used for generating.



Summary

- Here, we introduce the DistiLLM, addressing the problems of recent KD methods.
- **DistiLLM** includes:
- (1) **Skew KLD**, significantly improves optimization stability and generalizability.
- → **in-depth analysis for objective function**
- (2) **Adaptive off-policy**, comprises an adaptive SGO scheduler & off-policy strategy
- → **adaptive SGO**: alleviating potential **noisy feedback**
- → **off-policy strategy**: improving **sample efficient** of SGO
- → **better utilization of SGOs in KD**

Algorithm 1 Training pipeline of DistiLLM

Input: initial prob. ϕ, student q_{θ_0} with parameters θ_0, teacher p, total training iterations T, training & validation dataset $\mathcal{D}$, $\mathcal{D}_{val}$, empty replay buffer $\mathcal{D}_R$
Output: Student model q_{θ_T} with trained parameters θ_T
while $t \leq T$ **do**
    Randomly sample $u \sim \mathrm{Unif}(0,1)$
    / Linearly Decreasing Replay Ratio */*
    if $u < \lambda_R := \phi(1 - \frac{t}{T})$ **then**
        / Generate SGO & Update $\mathcal{D}_R$ */*
        Generate SGO $\{\tilde{\mathbf{y}}_i\}_{i=1}^B$ from $\{q_{\theta_t}(\cdot|\mathbf{x}_i)\}_{i=1}^B$
        Store SGO into $\mathcal{D}_R$; $\mathcal{D}_R \leftarrow \mathcal{D}_R \cup \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^B$
    end if
    if $u < \phi$ **then**
        / Use SGO in Off-policy Approach (Fig. 4(c)) */*
        Sample mini-batch $\{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^B$ from $\mathcal{D}_R$
    else
        / Use Sample from Fixed Dataset (Fig. 4(a)) */*
        Sample mini-batch $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^B$ from $\mathcal{D}$
    end if
    / Use S(R)KL */*
    Update θ_t by S(R)KL $D_{\mathrm{SKL}}^{(\alpha)}(\cdot,\cdot)$
    if do validation **then**
        $\mathcal{L}_{prev}, \phi \leftarrow$ `SGO_Scheduler`$(\mathcal{L}_{prev}, \mathcal{D}_{val}, q_{\theta_t})$
    end if
end while

/ Adaptive SGO Scheduler */*
def `SGO_Scheduler`$(\mathcal{L}_{\tilde{t}-1}, \mathcal{D}_{val}, q_\theta)$:
    / Compute Loss for Validation Set */*
    $\mathcal{L}_{\tilde{t}} \leftarrow \frac{1}{|D_{val}|}\sum_{\mathbf{x}_{val}, \mathbf{y}_{val}} \mathtt{Loss}(q_\theta, \mathbf{x}_{val}, \mathbf{y}_{val})$
    if $\mathcal{L}_{\tilde{t}} > \mathcal{L}_{\tilde{t}-1} + \varepsilon$ **then**
        Update $\phi_{\tilde{t}} \leftarrow \min(\phi_{\tilde{t}-1} + 1/N_\phi, 1.0)$
    else
        $\mathcal{L}_{\tilde{t}}, \phi_{\tilde{t}} \leftarrow \mathcal{L}_{\tilde{t}-1}, \phi_{\tilde{t}-1}$
    end if
    return $\mathcal{L}_{\tilde{t}}, \phi_{\tilde{t}}$

- **Instruction-following tasks**
- We trained all models on databricks-dolly-15k, open-source instruction-following dataset built by human.
- We evaluated all models on evaluation set of
- **databricks-dolly-15k** / **Self-instruct** / **Vicuna** / **Super-Natural instruction** / **Unnatural instruction**
- The metric we used are **ROUGE-L** / **GPT-4 feedback**

Skew KL Divergence

- Skewing KLD is highly effective with a more **favorable optimization process**.

$$D_{SKL}^{(\alpha)}(p, q_\theta) = D_{KL}(p, \alpha \cdot p + (1-\alpha) \cdot q_\theta)$$

- We can similarly define the α-SRKL by

$$D_{SRKL}^{(\alpha)}(p, q_\theta) = D_{KL}(q_\theta, (1-\alpha) \cdot p + \alpha \cdot q_\theta)$$

- We showed S(R)KL is superior to other loss functions, owing to its
- **More stable gradient** and **Smaller approximation error**

Stable Gradient

- We first analyze the gradients of KLD and Skew KLD to parameter θ.

$r_{p,q_\theta} = p(\boldsymbol{y}|\boldsymbol{x})/q_\theta(\boldsymbol{y}|\boldsymbol{x})$

$$\nabla_\theta D_{KL}(p, q_\theta) = -\boxed{r_{p,q_\theta}}\nabla_\theta q_\theta(\boldsymbol{y}|\boldsymbol{x})$$

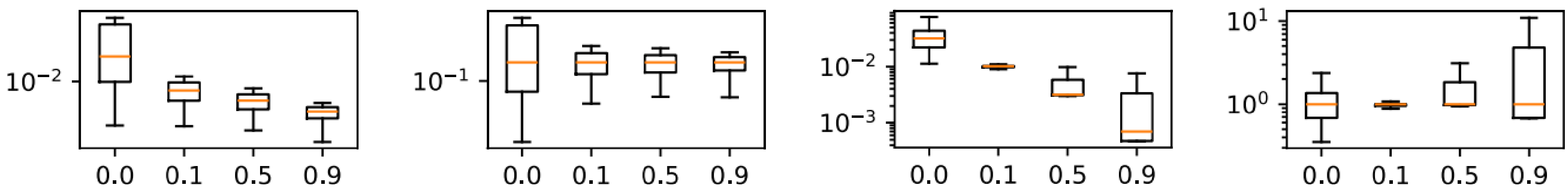
$$\nabla_\theta D_{SKL}^{(\alpha)}(p, q_\theta) = (1-\alpha) r_{p,\boxed{\tilde{q}_\theta}}\nabla_\theta q_\theta(\boldsymbol{y}|\boldsymbol{x})$$

$\tilde{q}_\theta(\boldsymbol{y}|\boldsymbol{x}) = \alpha p(\boldsymbol{y}|\boldsymbol{x}) + (1-\alpha) q_\theta(\boldsymbol{y}|\boldsymbol{x})$

- The gradient analysis for RKLD and Skew RKLD reveals similar trends.

$$\nabla_\theta D_{KL}(q_\theta, p) = -\left(\log r_{q_\theta,p} + 1\right)\nabla_\theta q_\theta(\boldsymbol{y}|\boldsymbol{x})$$

$$\nabla_\theta D_{SKL}^{(\alpha)}(q_\theta, p) = -\left(\log r_{q_\theta,\tilde{p}} + 1 - \alpha r_{q_\theta,\tilde{p}}\right)\nabla_\theta q_\theta(\boldsymbol{y}|\boldsymbol{x})$$



(a) SKL (w/o normalization) (b) SKL (w/ normalization) (c) SRKL (w/o normalization) (d) SRKL (w/ normalization)

Figure 9. Gradient coefficient distribution for SKL and SRKL across different skew values, α. Skewing KLD and RKLD effectively smooth the gradient norm, as seen in (a) and (c). For coefficients normalized by their median value, SKL shows a similar distribution when $\alpha > 0$ while SRKL exhibits explosion, as depicted in (b) and (d).

Small Approximation Error

- We showed that the empirical estimator of Skew KLD from mini-batch training has a **bounded L2 norm**.
- By achieving minimal error between the estimator and true divergence, we can
- Ensures **rapid convergence**,
- **High generalizability** by reflecting the full distribution from the empirical estimator.

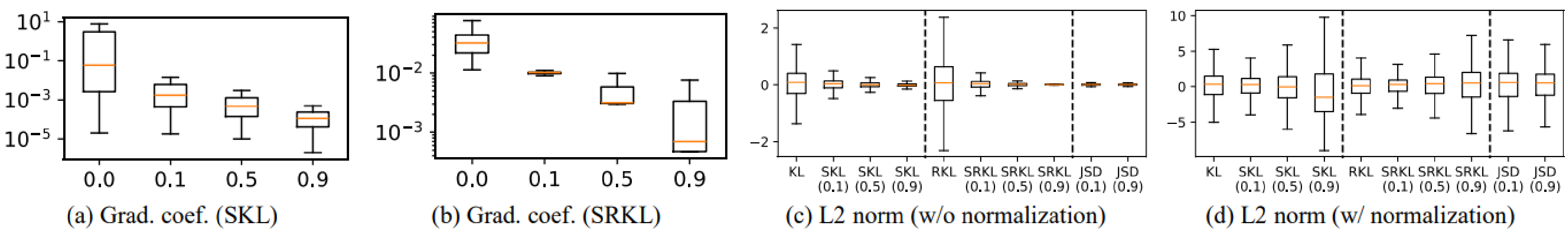


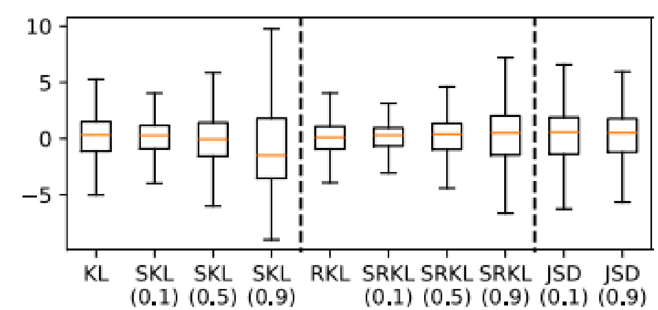
Figure 3. **(a)-(b)**: Gradient coefficient distribution for SKL and SRKL across different skew values α, as shown in Eq. 6–7. **(c)**: Distribution of differences between divergence values and their (exponential) moving average of α-S(R)KL, as shown in Thm. 1, and those of β-JSD by substituting SKL into JSD across different α and β, respectively. **(d)**: Normalized L2 norm distribution, dividing the L2 norm in **(c)** by corresponding gradient coefficient values.

- **Selecting α involves a trade-off:**
- The relationship between the upper bound of the normalized L2 norm and $\alpha \in [0, 1]$.
- Underscoring the importance of **balancing gradient and L2 norm scales**.
- Difference between SKL and JSD ($D_{JSD}^{(\beta)}(p, q_\theta) = \beta D_{SKL}^{(\beta)}(p, q_\theta) + (1-\beta) D_{SKL}^{(1-\beta)}(p, q_\theta)$):
- **SKL with a mild α** achieves a proper L2 norm value
- **JSD cannot simultaneously achieve moderate skew values** for both terms.

Remark 1. *By considering the reverse of approximated gradient scale, we have:*

$$\mathbb{E}[|\tfrac{1}{(1-\alpha)}(D_{SKL}^{(\alpha)}(p_n^1, p_n^2) - D_{SKL}^{(\alpha)}(p^1, p^2))|^2] \leq \frac{c_1^*(\alpha)}{n^2} + \frac{c_2 \log^2(\alpha n)}{(1-\alpha)^2 n} + \frac{c_3 \log^2(c_4 n)}{\alpha^2(1-\alpha)^2 n},$$

for $c_1^*(\alpha) = \min\left\{\frac{1}{\alpha^2(1-\alpha)^2}, \frac{\chi^2(p^1,p^2)^2}{(1-\alpha)^4}\right\}$.



(d) L2 norm (w/ normalization)

Comparison Between Objective Functions

- Conventional KLD, RKLD, JSD with a $\beta = 0.9$, and SKL and SRKL with a $\alpha = 0.1$

$$D_{JSD}^{(\beta)}(p, q_\theta) = \beta D_{KL}(p, \beta p + (1-\beta)q_\theta) + (1-\beta)D_{KL}(q_\theta, \beta p + (1-\beta)q_\theta)$$

- **(Left)** The results showed that our proposed objective function **generally outperform the others**.
- **(Right)** SKL and SRKL achieve remarkably high validation ROUGE-L for entire training phase, consistently showing **rapid convergence** and **strong generalization**.

Table 2. Evaluation of the effect of SKL and SRKL loss functions. **Bold** and underline indicate the best and second-best results, respectively, among those from the same evaluation dataset. We report the average and standard deviation of ROUGE-L scores across five random seeds.

Loss Function	Dolly Eval	Self-Instruct	Vicuna Eval	Super-Natural	Unnatural
KLD	23.52 (0.22)	11.23 (0.46)	15.92 (0.41)	20.68 (0.16)	23.38 (0.13)
RKLD	23.82 (0.34)	10.90 (0.58)	<u>16.11 (0.46)</u>	22.47 (0.21)	23.03 (0.11)
Generalized JSD	24.34 (0.35)	12.01 (0.54)	15.21 (0.61)	25.08 (0.36)	27.54 (0.07)
SKL	<u>24.80 (0.12)</u>	<u>12.86 (0.34)</u>	**16.20 (0.57)**	**26.26 (0.41)**	<u>28.06 (0.08)</u>
SRKL	**25.21 (0.27)**	**12.98 (0.24)**	15.77 (0.39)	<u>25.83 (0.15)</u>	**28.62 (0.10)**

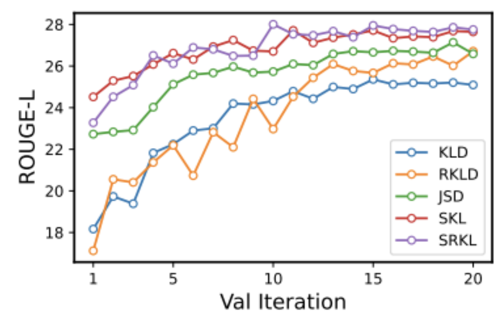
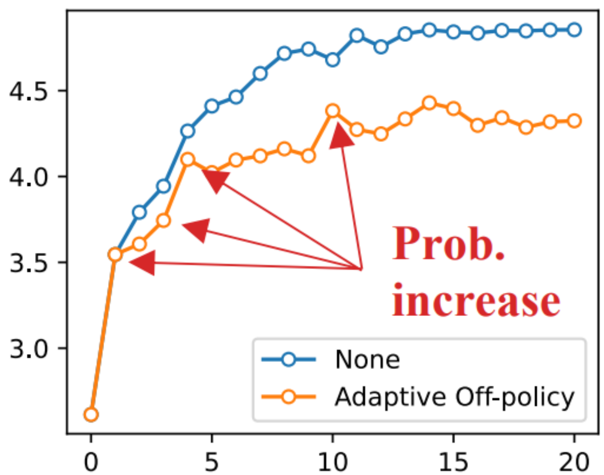


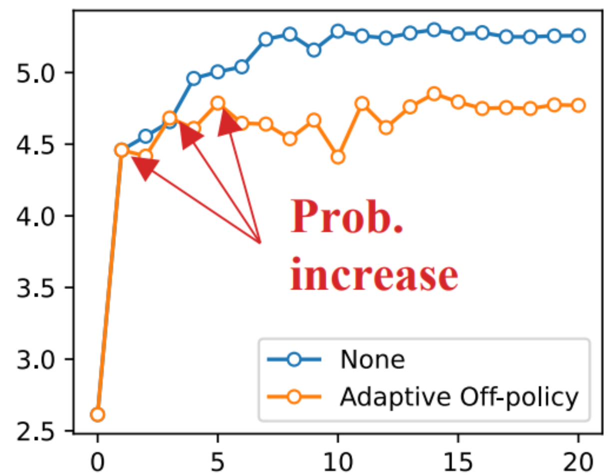
Figure 6. ROUGE-L scores for the validation set across the different loss functions.

Adaptive SGO Scheduler

- Effectively balance between noisy feedback and training-inference mismatch
- We define the probability of using SGOs, denoted as ϕ.
- Our scheduler **starts with low ϕ value**, **gradually increasing** during training.
- We primarily rely on **validation loss** as a metric.
- We adjust ϕ by **comparing the current and previous validation losses**; an increase in validation loss leads to an increase in ϕ.



(a) SKL

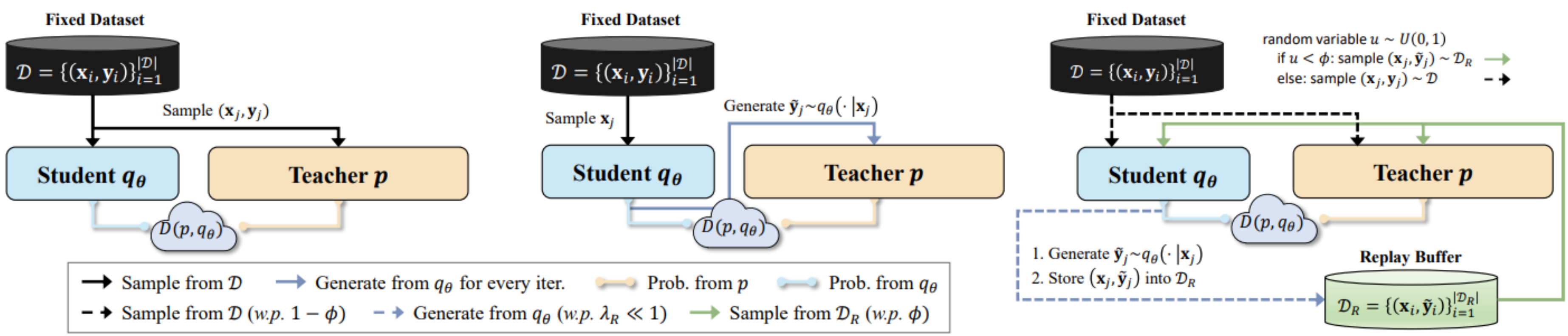


(b) SRKL

Off-policy Approach for Sample Efficiency

Off-policy Approach

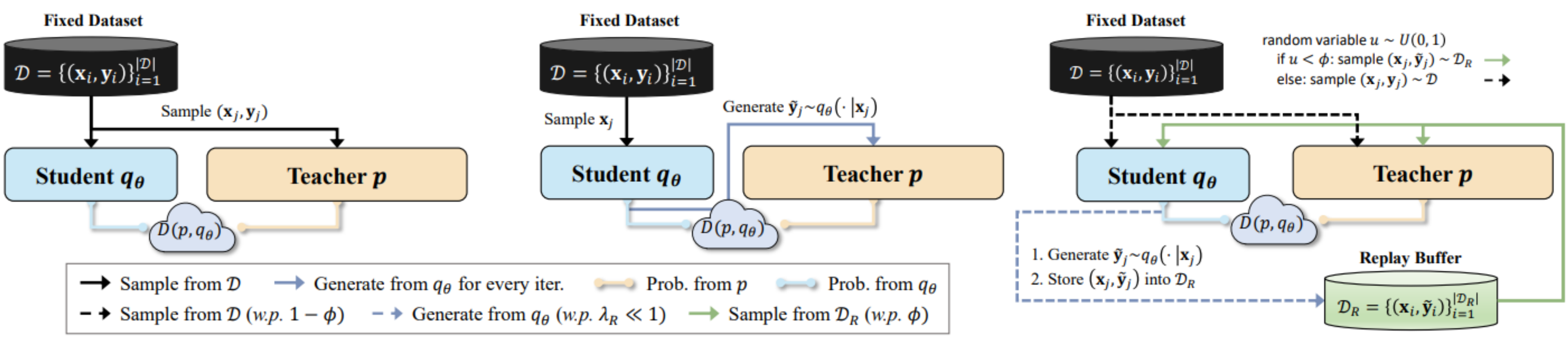
- On-policy distillation is computationally heavy, generating SGO for every iteration.
- Off-policy approach can improve computational efficiency of distillation.
- Motivated from off-policy RL, we store SGOs into replay buffer.
- We utilize these samples for multiple times, instead of disposable SGO of on-policy.



(a) KD from Fixed Dataset (b) On-policy Approach (c) Adaptive Off-policy Approach (Ours)

High Bias Error of Off-policy RL

- Off-policy RL is prone to high bias error, when there is a significant difference between past and current policies.
- **Early training**: student model parameters rapidly evolve → **focusing on using current SGOs** with a small replay ratio
- **Late training**: student model converge → **highly reusing stored SGOs** with a high replay ratio



(a) KD from Fixed Dataset (b) On-policy Approach (c) Adaptive Off-policy Approach (Ours)

Synergy Between Skew KLD and Off-policy Approach

- The success of off-policy approach **stems from the fast convergence of S(R)KL** while other loss functions cannot be achieved.
- Both SKL and SRKL have a significant **early-stage improvement**, effectively leveraging the off-policy without high bias issues.
- Unlike other loss functions (KLD, JSD) that suffer performance drops when switching from on-policy to off-policy, **our method maintains its efficacy**.

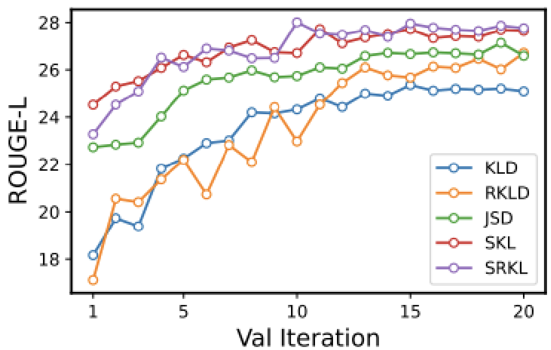


Figure 6. ROUGE-L scores for the validation set across the different loss functions.

Table 4. Application of our off-policy method to the existing KD methods. Off-policy significantly reduces the performance of ImitKD and GKD, as opposed to our proposed **DistiLLM**.

Dataset	Dolly Eval		Self-Instruct		Super-Natural	
Sampling	on-	off-	on-	off-	on-	off-
ImitKD (Lin et al., 2020)	21.63	20.62	10.85	10.09	19.94	18.04
GKD (Agarwal et al., 2024)	23.75	22.89	12.73	12.78	26.05	24.97
DistiLLM (ours)	26.37	26.12	13.14	13.16	28.24	28.20

Comparison Between Sampling Methods

- Mixed strategy: using on-policy approach *w.p.* 0.5
- Adaptive SGO scheduler **effectively balances** the trade-off between the **risk of noisy feedback and training-inference mismatch**.
- Our off-policy approach achieves **$2.2\times$ to $3.4\times$ faster training speed** compared to the on-policy or mixed strategy.

Table 3. Evaluation of the adaptive off-policy approach. We apply SKL and SRKL with all generation methods. We report the average and standard deviation of ROUGE-L scores across five random seeds. The best and second best performances are highlighted **bold** and underline.

Generation	Dolly Eval	Self-Instruct	Vicuna Eval	Super-Natural	Unnatural
Skew KLD	24.80 (0.12)	12.86 (0.34)	16.20 (0.57)	26.26 (0.41)	28.06 (0.08)
∟ On-policy	24.27 (0.46)	<u>13.13 (0.44)</u>	16.39 (0.21)	25.87 (0.18)	26.49 (0.09)
∟ Mixed	25.27 (0.35)	12.24 (0.69)	17.19 (0.29)	25.30 (0.33)	26.51 (0.11)
∟ Adaptive (ours)	**25.90 (0.20)**	**13.24 (0.30)**	**17.59 (0.44)**	**27.62 (0.05)**	**28.30 (0.11)**
+ Off-policy (ours)	<u>25.79 (0.28)</u>	13.03 (0.29)	<u>17.41 (0.15)</u>	<u>27.32 (0.09)</u>	<u>28.13 (0.21)</u>
Skew RKLD	25.21 (0.27)	12.98 (0.24)	15.77 (0.39)	25.83 (0.15)	28.62 (0.10)
∟ On-policy	26.04 (0.33)	12.93 (0.54)	17.45 (0.37)	27.29 (0.12)	28.72 (0.10)
∟ Mixed	26.01 (0.61)	12.24 (0.69)	17.19 (0.29)	26.40 (0.34)	29.02 (0.14)
∟ Adaptive (ours)	**26.37 (0.21)**	<u>13.14 (0.37)</u>	<u>18.32 (0.17)</u>	**28.24 (0.22)**	**30.11 (0.04)**
+ Off-policy (ours)	<u>26.11 (0.68)</u>	**13.14 (0.69)**	**18.46 (0.53)**	<u>27.51 (0.03)</u>	<u>29.35 (0.07)</u>

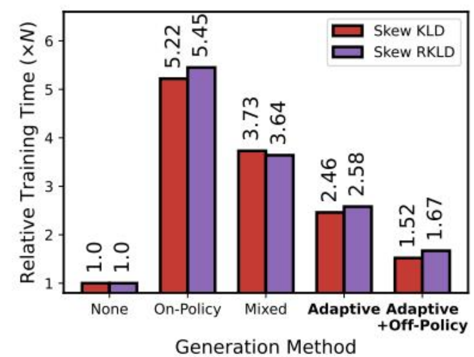
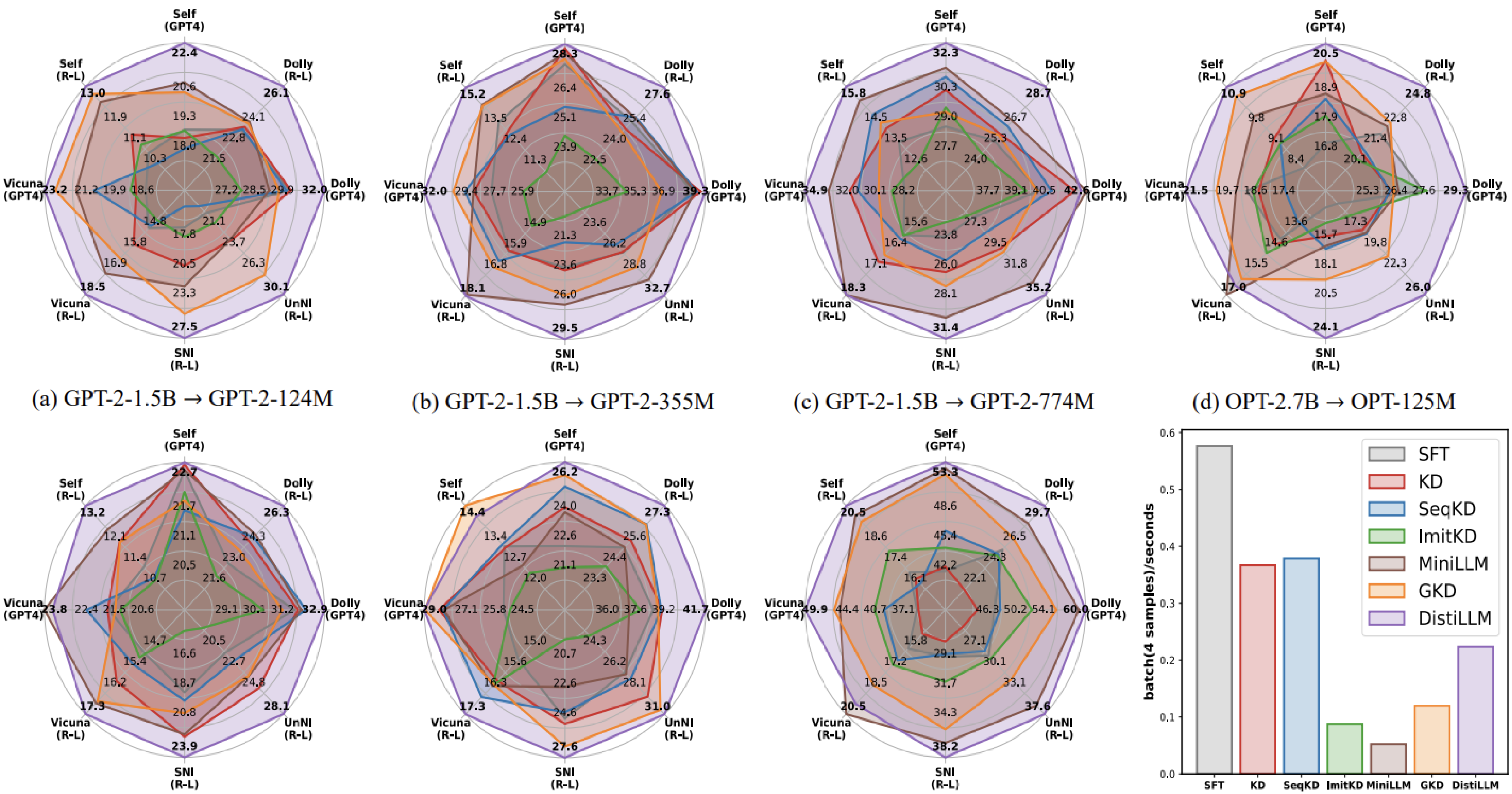


Figure 7. Relative training time for different generation methods for skew KLD and skew RKLD. The adaptive off-policy approach shows significant efficiency.

Main Results

- DistiLLM **outperforms** other baselines for **ROUGE**, **GPT4**, and **training speedup**.



(a) GPT-2-1.5B → GPT-2-124M (b) GPT-2-1.5B → GPT-2-355M (c) GPT-2-1.5B → GPT-2-774M (d) OPT-2.7B → OPT-125M

(e) OPT-2.7B → OPT-350M (f) OPT-2.7B → OPT-1.3B (g) OLLaMA2-7B → OLLaMA2-3B (h) Training Speed (OLLaMA2-3B)