DistiLLM-2: A Contrastive Approach Boosts the Distillation of LLMs

(ICML 2025 Oral Spotlight)

fzeng 05/08/2025

Why distillation?

- Smaller language models (SLMs) more efficient, easier to serve
- Used as draft models in speculative decoding
- Standard objective: minimize KL between teacher and student distribution (same as cross entropy up to a constant that only depends on the teacher)
- Why does this help?
    - Soft labels: teaches weighing of relative options
    - Ha

Why does distillation help?

mapping from input vectors to output vectors. For cumbersome models that learn to discriminate between a large number of classes, the normal training objective is to maximize the average log probability of the correct answer, but a side-effect of the learning is that the trained model assigns probabilities to all of the incorrect answers and even when these probabilities are very small, some of them are much larger than others. The relative probabilities of incorrect answers tell us a lot about how the cumbersome model tends to generalize. An image of a BMW, for example, may only have a very small chance of being mistaken for a garbage truck, but that mistake is still many times more probable than mistaking it for a carrot.

Distilling the Knowledge in a Neural Network (Hinton et al, 2015)

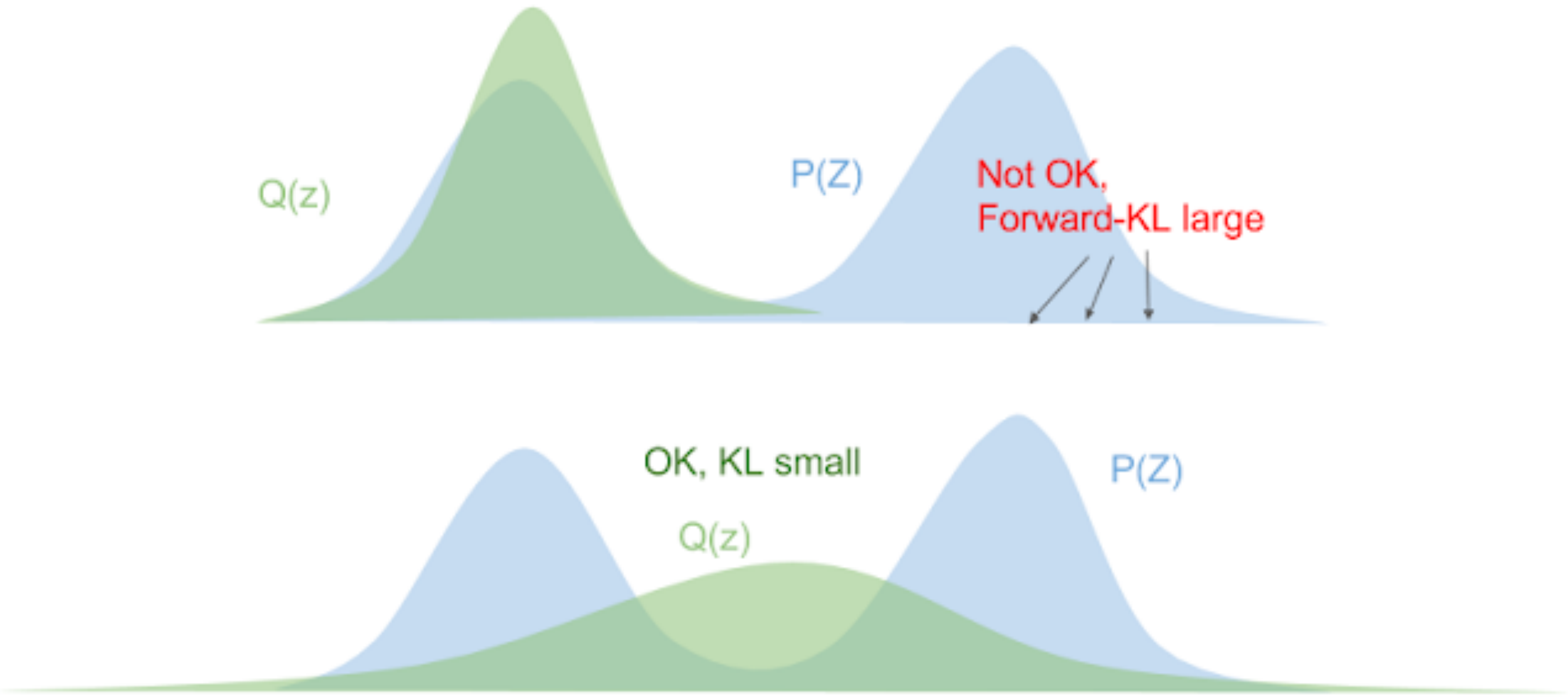
KL Primer

- Information-theoretic view: how "wasteful" is it to encode data that is drawn from distribution p using a codebook that is optimized for distribution q?
- $$\mathrm{KL}\left(p \,\|\, q_\theta\right) = \sum_{x \in \mathcal{X}} p(x) \, \log \frac{p(x)}{q_\theta(x)}$$
- Some properties: not symmetric, not a metric, $\mathrm{KL}(p \,\|\, p) = 0$

Problems with KL for distillation

- KL is zero-avoiding:

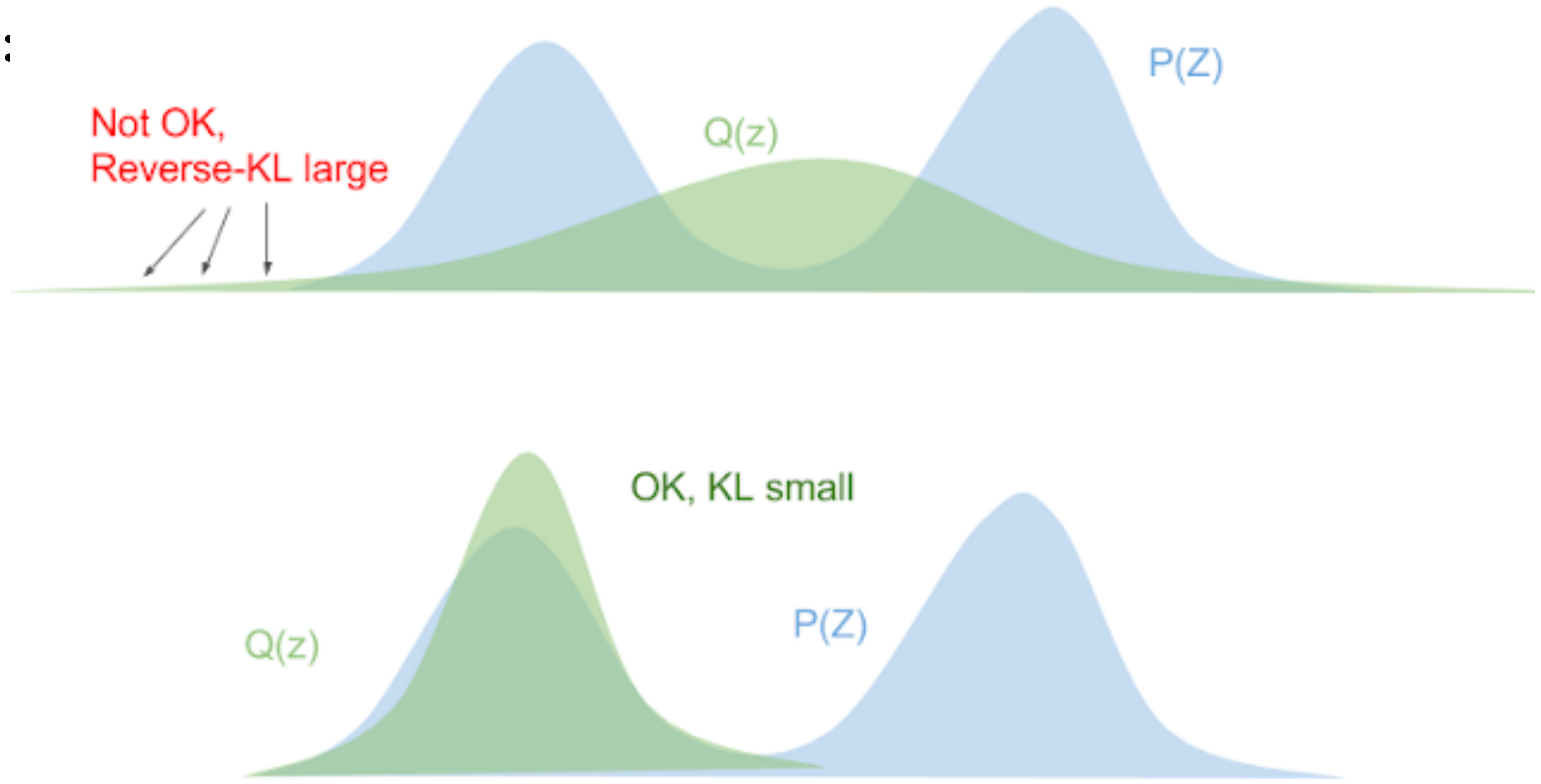
$$\mathrm{KL}\left(p \parallel q_\theta\right) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q_\theta(x)}$$



Problems with KL for distillation

$$\mathrm{KL}\left(p \,\|\, q_\theta\right) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q_\theta(x)}$$

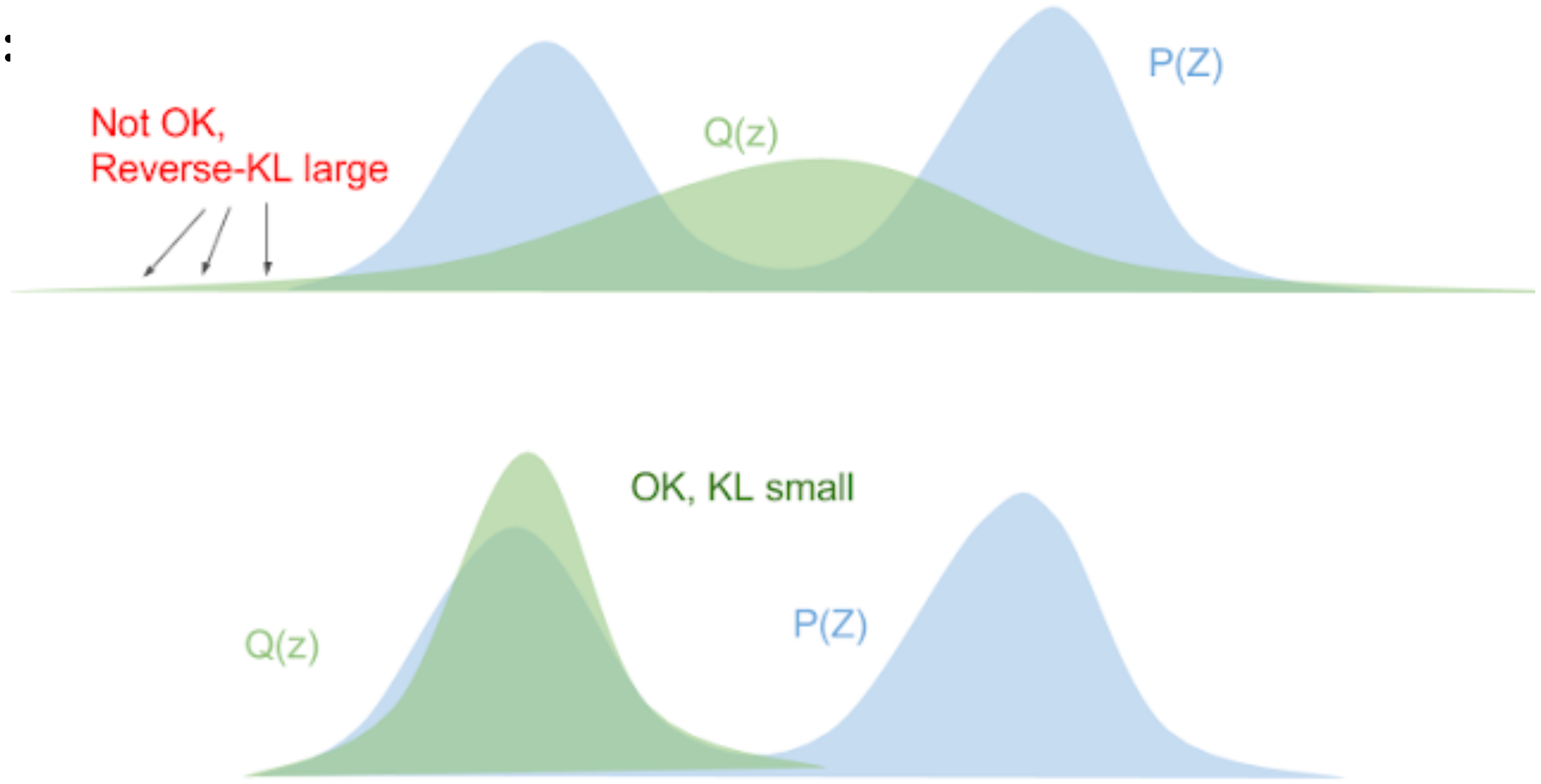
- What about reverse KL: $\mathrm{KL}(q_\theta \,\|\, p)$?
- It becomes zero-forcing:



Problems with KL for distillation

$$\mathrm{KL}\left(p \,\|\, q_\theta\right) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q_\theta(x)}$$

- What about reverse KL: $\mathrm{KL}(q_\theta \,\|\, p)$?
- It becomes zero-forcing:



Problems with KL for distillation

$$\mathrm{KL}\big(p \parallel q_\theta\big) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q_\theta(x)}$$

- Large noisy gradients:

$$\nabla_\theta \mathrm{KL}\left(p, q_\theta\right) = -\frac{p(x)}{q_\theta(x)} \nabla_\theta q_\theta(x)$$

- Blows up if student assigns low probability to sample

Prior work: DistiLLM and Skew-KL (ICML 2024)

- Skew KL: interpolate the target between teacher and student
- $D_{\mathrm{SKL}}^{(\alpha)}\left(p, q_\theta\right) = D_{\mathrm{KL}}\left(p, \alpha p + (1-\alpha) q_\theta\right)$
- More stable gradient updates
- Faster convergence, better performance

DistiLLM-2

- DistiLLM only focused on loss formulation (with SKL/RSKL)
- However data curation is also important:
  - Have student learn from teacher-generated outputs (TGO)?
  - Have teacher correct student-generated outputs (SGO)?
- DistiLLM-2: consider both loss formulation and data curation

Contrastive Loss

- If you have TGO and SGO, one approach can be: encourage teacher outputs and discourage student outputs
- Recall DPO: increase winning response, decrease losing response

$$-\log \sigma \Bigg(\underbrace{\lambda \log \frac{q_\theta(\boldsymbol{y}_w|\boldsymbol{x})}{q_{\text{ref}}(\boldsymbol{y}_w|\boldsymbol{x})}}_{\text{increase } q_\theta(\boldsymbol{y}_w|\boldsymbol{x})} - \underbrace{\lambda \log \frac{q_\theta(\boldsymbol{y}_l|\boldsymbol{x})}{q_{\text{ref}}(\boldsymbol{y}_l|\boldsymbol{x})}}_{\text{decrease } q_\theta(\boldsymbol{y}_l|\boldsymbol{x})} \Bigg),$$

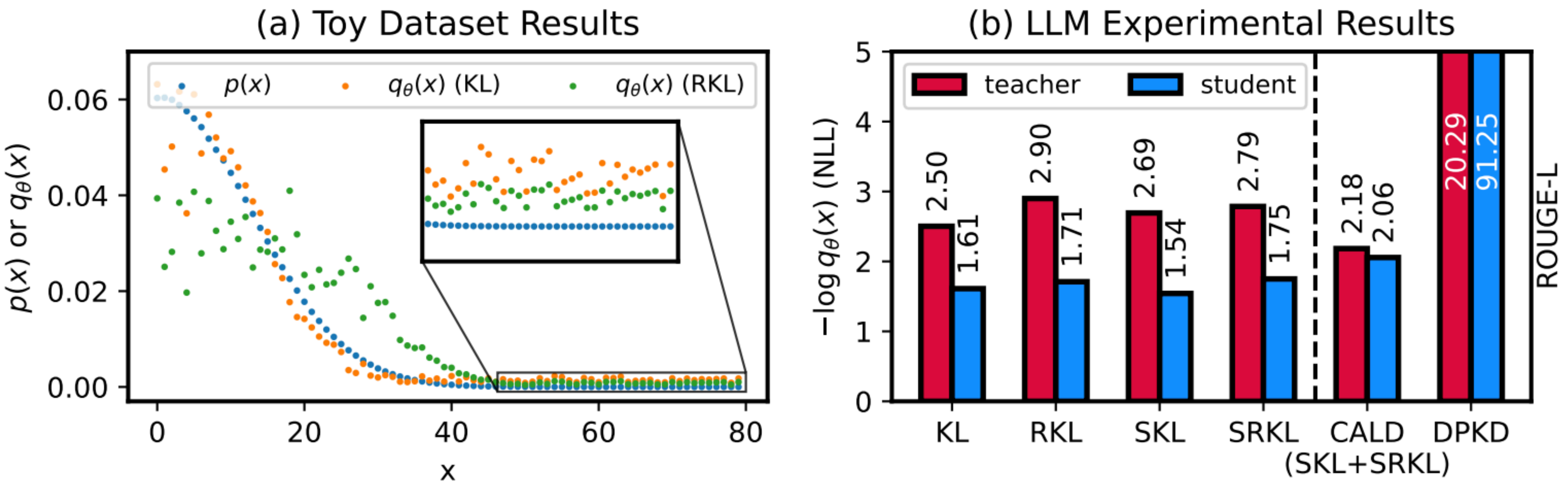
Contrastive Loss

- What if we just apply DPO idea directly? (Direct Preference Knowledge Distillation for Large Language Models)

$$-\log\sigma\ \underbrace{\left(\lambda\log\frac{q_\theta(\boldsymbol{y}_t|\boldsymbol{x})}{p(\boldsymbol{y}_t|\boldsymbol{x})}-\lambda\log\frac{q_\theta(\boldsymbol{y}_s|\boldsymbol{x})}{p(\boldsymbol{y}_s|\boldsymbol{x})}\right)}_{\text{inherently small } p(\boldsymbol{y}_s|\boldsymbol{x})\ \rightarrow\ \text{overly decrease } q_\theta(\boldsymbol{y}_s|\boldsymbol{x})},$$

- Reward hacking possible: increase of encouraging the teacher outputs, can just decrease probability of student outputs

Intuition



- Can indeed see push-up effect on training with KL (covering all modes) and push-down effect of training with RKL (zero forcing)

Contrastive Approach for LLM Distillation (CALD)

$$\mathcal{L}_{\text{CALD}} = \frac{1}{2|\mathcal{D}|} \sum_{(\boldsymbol{x}, \boldsymbol{y}_t, \boldsymbol{y}_s) \sim \mathcal{D}} D_{\text{SKL}}^{(\alpha)}(\boldsymbol{x}, \boldsymbol{y}_t) + D_{\text{SRKL}}^{(\alpha)}(\boldsymbol{x}, \boldsymbol{y}_s).$$

- In previous work, found that using the same response type for SKL and SRKL didn't help
- Idea: encourage desirable behavior from teacher, further suppress already low probability behavior from student
- They show this can be re-written in a form that looks similar to DPO loss
    - But without reward hacking problems as denominators are interpolated between student and teacher distribution

Curriculum for α

- Interpolation factor α controls "speed" of learning of student: q_θ vs $\alpha p + (1 - \alpha)q_\theta$
- Large α: stable training but model doesn't learn as much
- Small α: less stable training and slower convergence, but can get closer to teacher

Curriculum for α

- But really: amount to update should depend on how “hard” the sample is
- “Easy” samples: choose small α, and vice versa
- Set α for each sample such that the likelihood of teacher/(α- interpolated student + teacher) is constant across samples

Results

- Small improvements over baselines
- Still a big gap remaining?
- Frontier labs seem to be getting small models right

Table 3. Comparison results on the GSM8k and MATH benchmarks. The best *pass@1* score is highlighted in **bold**.

Method	Qwen2-Math-7B-Inst ($\mathcal{M}_T$) → Qwen2-Math-1.5B ($\mathcal{M}_S$) GSM8K Pass@1	MATH Pass@1	AVG. Pass@1	Qwen2.5-Math-7B-Inst ($\mathcal{M}_T$) → Qwen2.5-Math-1.5B ($\mathcal{M}_S$) GSM8K Pass@1	MATH Pass@1	AVG. Pass@1
$\mathcal{M}_T$	83.93	41.28	62.61	89.31	44.82	67.07
$\mathcal{M}_S$	74.53	25.56	50.05	77.33	27.14	52.24
GKD	75.44	34.16	54.80	80.21	40.54	60.38
DistiLLM	75.59	34.54	55.07	81.05	41.14	61.10
DistiLLM-2	**76.27**	**35.58**	**55.93**	**81.20**	**42.94**	**62.07**

Table 4. Comparison results on the HumanEval (HEval) and MBPP benchmarks. The best *pass@1* score is highlighted in **bold**.

Method	DS-Coder-6.9B-Inst ($\mathcal{M}_T$) → DS-Coder-1.3B ($\mathcal{M}_S$) HEval Pass@1	MBPP Pass@1	AVG. Pass@1	Qwen2.5-Coder-7B-Inst ($\mathcal{M}_T$) → Qwen2.5-Coder-1.5B ($\mathcal{M}_S$) HEval Pass@1	MBPP Pass@1	AVG. Pass@1
$\mathcal{M}_T$	85.37	82.54	83.96	75.61	74.60	75.61
$\mathcal{M}_S$	50.61	72.22	61.42	30.73	60.84	45.79
GKD	54.88	74.34	64.61	40.85	61.90	51.38
DistiLLM	53.65	74.34	64.00	39.63	62.17	50.90
DistiLLM-2	**59.92**	**75.66**	**67.79**	**42.24**	**62.70**	**52.47**

Algorithm

Algorithm 1 Training pipeline of DistiLLM-2

Input: training iterations T, initial skew coefficient α_0, teacher p, student q_{θ_0} with parameter θ_0, prompt set

Output: Student model q_{θ_E} with trained parameters θ_E

for epoch $e = 1, 2, \ldots, E$ **do**

    /* Sample *batched on-policy* responses */

    Sample responses $\boldsymbol{y}_t, \boldsymbol{y}_s$ from teacher $p(\cdot|\boldsymbol{x})$ and student $q_{\theta_{e-1}}(\cdot|\boldsymbol{x})$ for given prompt $\boldsymbol{x}$

    Construct $\mathcal{D}_t = \{(\boldsymbol{x}, \boldsymbol{y}_t, \boldsymbol{y}_s)\}$ for training dataset for training epoch e.

    Initialize $\theta_e \leftarrow \theta_{e-1}$

    for iteration $\tau = 1, 2, \ldots, T$ **do**

        Sample mini-batch: $\mathcal{B} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}_t^{(i)}, \boldsymbol{y}_s^{(i)})\}_{i=1}^{|\mathcal{B}|}$ from $\mathcal{D}_t$

        /* Curriculum-based adaptive update for α */

        Update $\alpha_t \leftarrow 1 - (1 - \alpha_0) \cdot \frac{m}{p(\boldsymbol{y}_s|\boldsymbol{x}) - q_\theta(\boldsymbol{y}_s|\boldsymbol{x})}$ and $\alpha_s \leftarrow 1 - (1 - \alpha_0) \cdot \frac{m}{p(\boldsymbol{y}_t|\boldsymbol{x}) - q_\theta(\boldsymbol{y}_t|\boldsymbol{x})}$

        /* Gradual increasing coefficient for SRKL */

        Update $\beta \leftarrow \texttt{clip}(\frac{e}{E} + \frac{\tau}{T}, \beta_0, 1)$

        /* Improved contrastive loss function (§3.3)*/

        Update θ_e by minimizing $\mathcal{L}_{\text{DistiLLM-2}} = \frac{1}{2B} \sum \left[(1-\beta) D_{\text{SKL}}^{(\alpha_t)}(\boldsymbol{x}, \boldsymbol{y}_t) + \beta D_{\text{SRKL}}^{(\alpha_s)}(\boldsymbol{x}, \boldsymbol{y}_s)\right]$

    end for

end for