Domain Mixture Design via Log-Likelihood Differences for Aligning Language Models with a Target Model

Ryo Kishino[1], Riku Shiomi[1], Hiroaki Yamagiwa[2], Momose Oyama[1,3], Hidetoshi Shimodaira[1,3]
[1]Kyoto University [2]SB Intuitions [3]RIKEN

EMNLP2026 BUDAPEST

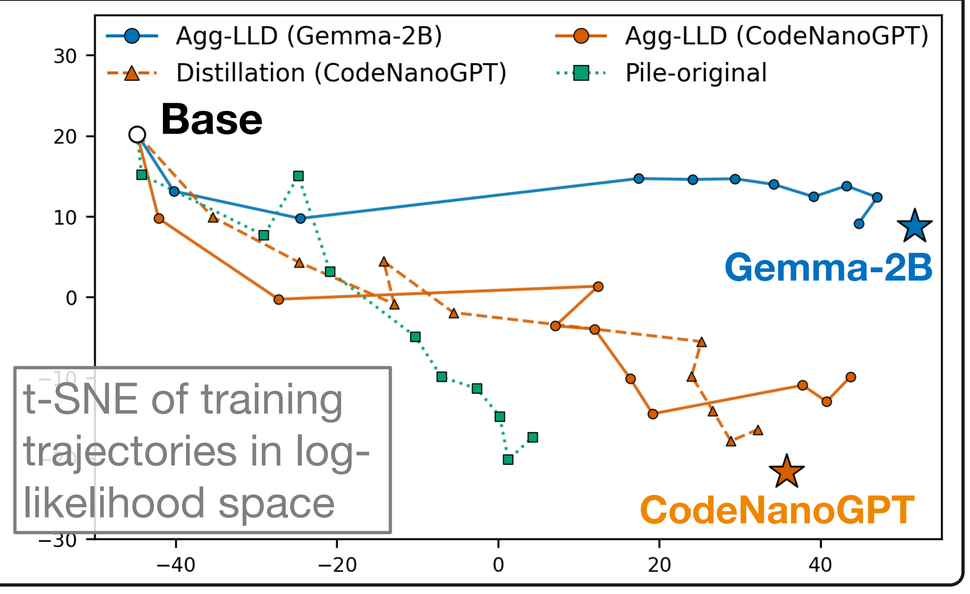
Pull a base LM toward a target LM by designing only the domain mixture — no teacher outputs.

Problem

- Bring a **base** LM closeer to a **target** LM in distribution by **only designing the domain mixture** of its (continued) pretraining data.
- Distillation requires teacher outputs during training.

Idea

- Models are **points in log-likelihood space**.
- Design domain weights so the **update direction** points to the target: softmax of domain-level **log-likelihood differences (LLD)**.



Log-Likelihood Space

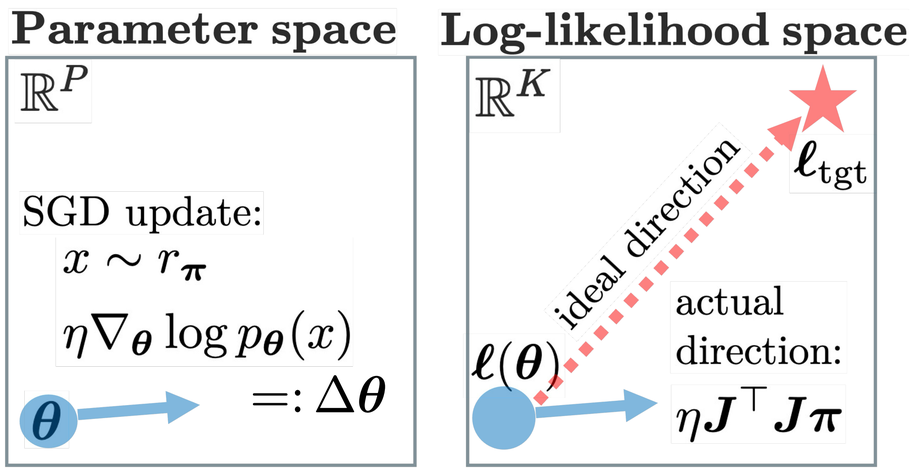
- Let $\mathcal{D}_k$ be evaluation texts from the Pile domain k.
- Represent a model p_θ as a point in $\mathbb{R}^K$ using domain-wise mean log-likelihoods [Oyama+ 2025]

$$\ell_k(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}_k|} \sum_{x \in \mathcal{D}_k} \log p_{\boldsymbol{\theta}}(x), \quad \boldsymbol{\ell}(\boldsymbol{\theta}) = (\ell_1, \ldots, \ell_K)^\top$$

- Log-likelihood vectors give a model map with

$$2\mathrm{KL}(p_i, p_j) \approx \|\boldsymbol{\ell}_i - \boldsymbol{\ell}_j\|^2 / N$$

Method



❶ Update direction under mixture π

SGD with a text $x \sim r_{\boldsymbol{\pi}} = \Sigma_k \pi_k\, r_k$ moves $\ell(\boldsymbol{\theta})$ by

$$\mathbb{E}_{x \sim r_{\boldsymbol{\pi}}}[\ell(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}) - \ell(\boldsymbol{\theta})] \approx \eta \boldsymbol{J}^\top \boldsymbol{J} \boldsymbol{\pi}, \quad \boldsymbol{J} = \nabla \ell(\boldsymbol{\theta})$$

❷ Align it with the target direction

$$\tilde{\boldsymbol{\pi}} = (\boldsymbol{J}^\top \boldsymbol{J})^{-1} (\boldsymbol{\ell}_{\mathrm{tgt}} - \boldsymbol{\ell}),$$

$$\max_{\boldsymbol{\pi} \in \Delta^{K-1}} \boldsymbol{\pi}^\top \tilde{\boldsymbol{\pi}} - \tau \mathrm{KL}(\boldsymbol{\pi}, \boldsymbol{\pi}') \Longrightarrow \pi_k \propto \pi_k' \exp(\tilde{\pi}_k / \tau)$$

$\boldsymbol{\pi}'$: reference weights (Pile-original)

❸ LLD rule: drop $(\boldsymbol{J}^\top \boldsymbol{J})^{-1}$

$$\pi_k \propto \pi_k' \exp\left(\frac{\ell_{\mathrm{tgt},k} - \ell_k}{\tau}\right)$$

❹ Fix the recipe in two passes

Pass 1
Iterative-LLD: re-estimate $\pi^{(t)}$ at steps $t \in \mathcal{T}$

→ **Aggregate**
geometric mean $\pi_k^* \propto \left(\prod_{t \in \mathcal{T}} \pi_k^{(t)}\right)^{1/|\mathcal{T}|}$

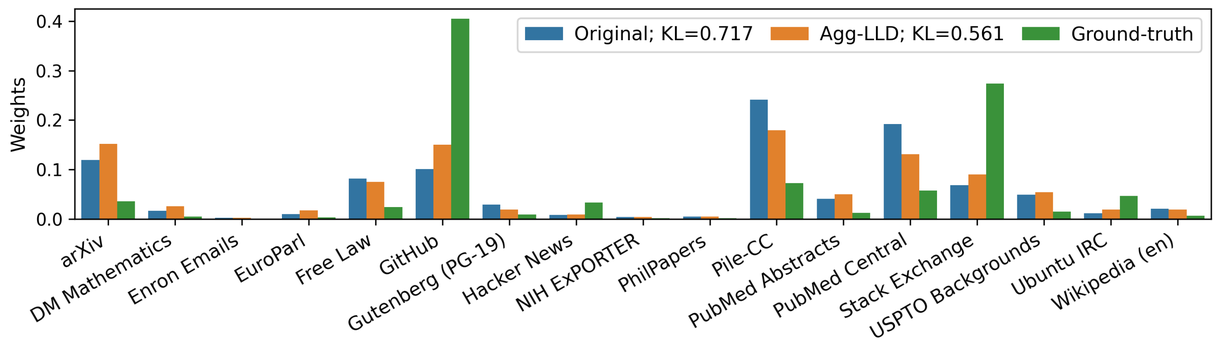
→ **Pass 2**
Aggregated-LLD: train with **fixed** π^*, no target access

$\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}} \sum_t \mathrm{KL}(\boldsymbol{\pi}, \boldsymbol{\pi}^{(t)})$. Fixed recipe: auditable & reproducible.

Results

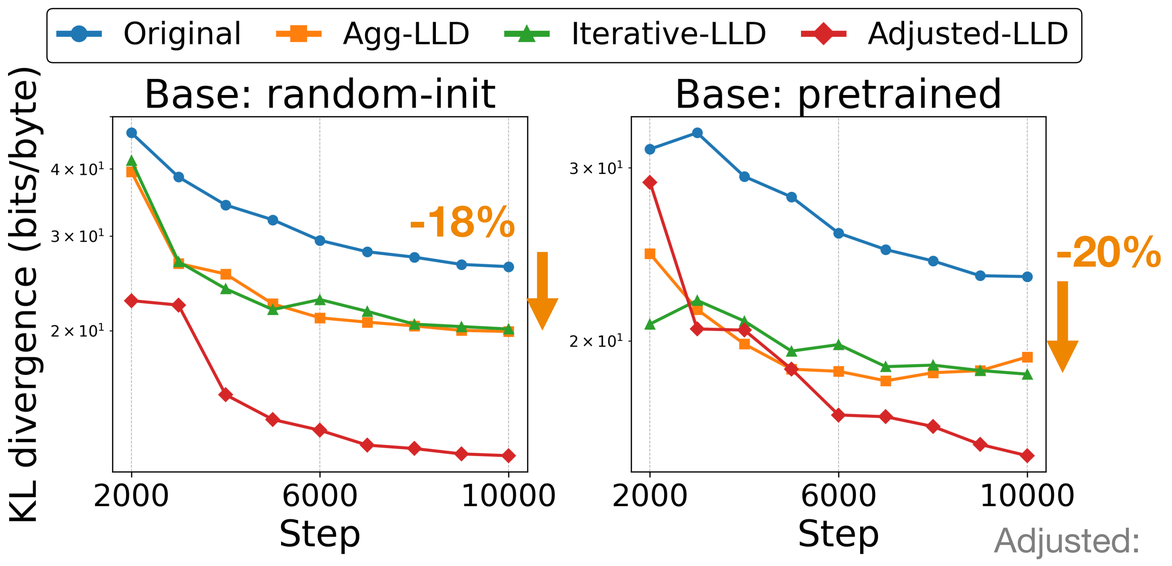
(i) Recovers the target weighting pattern

- Target: CodeNanoGPT (code-related domain ×4)
- Base: random-init
- KL to ground-truth weights: 0.717→0.561



(ii) Aligns with public LLMs

- Target: Gemma-2B. KL to the target during training
- Final KL (rand/pre): 17.9/12.8 → 14.6/10.3 [bpb]



(iii) Close to knowledge distillation

- Target: CodeNanoGPT. KL to the target.
- Final KL (rand/pre): **3.42/2.84** vs. distill w/ CE 3.33/2.40

