NEURIPS 2026

# CausalMix

## Data Mixture as Causal Inference for Language Model Training

Zinan Tang[1,2*], Yukun Zhang[2*], Shaomian Zheng[2], Zhuoshi Pan[1], Qizhi Pei[3], Dingnan Jin[2], Jun Zhou[2†], Yujun Wang[1†], Biqing Huang[1†]

[1]Tsinghua University [2]Ant Group [3]Renmin University of China [*]Equal contribution [†]Corresponding author

zinantang.pages.dev/causalmix

tzn26@mails.tsinghua.edu.cn · hbq@tsinghua.edu.cn

# Motivation

Proxy-model mixers assume the data distribution stays fixed

### One global map

RegMix-style methods fit a single map from mixture weights to validation loss.

### Refit on every shift

When the data pool shifts, the map no longer holds and proxy runs start from scratch.

### Loss vs. downstream

Lower training loss does not necessarily mean better downstream performance.

A single mixture need not be optimal across different data states.

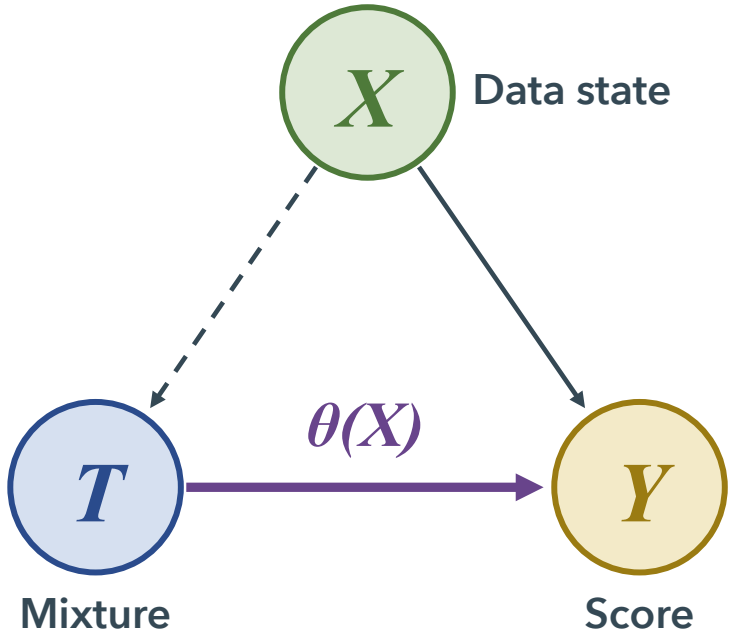
KEY IDEA

# “How does a relative change in domain proportions causally affect downstream performance under the current data state?”

| | | |
|---|---|---|
| $X$ | **Covariates** | complexity, difficulty and quality of the pool |
| $T$ | **Treatment** | the domain mixture of a training run |
| $Y$ | **Outcome** | downstream performance after training |

# Causal Formulation

Mixture as treatment, data state as covariates




Treatment: the log-mixture over $K$ domains

$$T = (T_1, \ldots, T_K) \in \Delta^{K-1}, \qquad Z = \log(T + \varepsilon)$$

Partially linear response; $\theta_0(x)$ is the state-conditioned marginal data return, a CATE

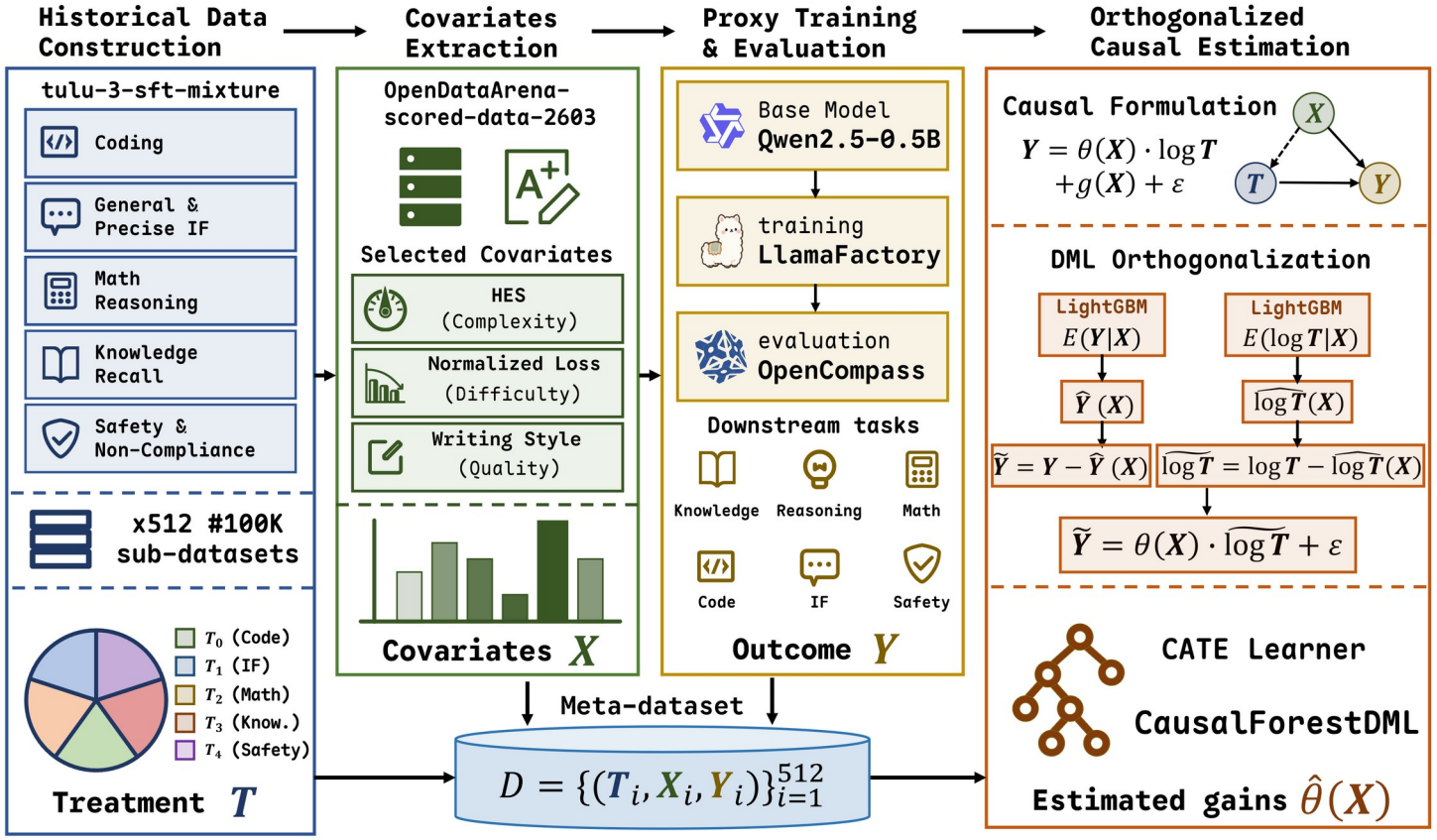
$$\mu(x, Z) = \mathbb{E}[Y(t) \mid X = x] \;\approx\; g(x) + \theta_0(x)^\top Z$$

Mixtures are drawn independently of the pool, so $\theta_0(x)$ is identifiable:

$$Y(t) \perp T \mid X \implies \mathbb{E}[Y \mid X = x,\, Z = z] = \mu(x, z)$$

# CausalMix Pipeline

512 proxy runs become a meta-dataset of $(T, X, Y)$ triplets




### 01 Historical data

512 pools × 100K examples from tulu-3-sft-mixture, 5 domains

### 02 Covariates $X$

HES, Normalized_Loss and Writing_Style from OpenDataArena

### 03 Outcome $Y$

Qwen2.5-0.5B proxy, scored on six capability groups

### 04 Causal estimation

LightGBM nuisances + CausalForestDML estimate $\theta(X)$

# Orthogonal Estimation

Residualize on the data state, then learn heterogeneous returns

## Double Machine Learning

Cross-fitted LightGBM residualizes $Y$ and $Z$ on $X$:

$$\widetilde{Y} = Y - \mathbb{E}[Y \mid X], \qquad \widetilde{Z} = Z - \mathbb{E}[Z \mid X]$$

CausalForestDML then minimizes the orthogonal R-loss:

$$\hat{\theta} = \arg\min_{\theta} \sum_{i} \left( \widetilde{Y}_i - \theta(X_i)^\top \widetilde{Z}_i \right)^2$$

## From returns to a mixture

**CausalMix-A** normalizes the positive returns:

$$T_k^{\mathrm{A}} = \frac{[\hat{\theta}_k(X_{\mathrm{tar}})]_+}{\sum_{j=1}^{K} [\hat{\theta}_j(X_{\mathrm{tar}})]_+}$$

**CausalMix-S** averages the top 100 of 100K Dirichlet candidates:

$$T^{\mathrm{S}} = \frac{1}{100} \sum_{m \in \text{Top-100}} T^{(m)}$$

# Experimental Setup

One meta-dataset of proxy runs, many targets

### Data

tulu-3-sft-mixture with 5 domains; 512 pools × 100K examples

### Covariates

HES (complexity), Normalized_Loss (difficulty), Writing_Style (quality)

### Evaluation

Six capability groups via OpenCompass; dev set gives $Y$, unseen set held out

### Baselines

Equal, Grid, RegMix, DoReMi, ODM and DMO on the same runs

# Main Results

Best development-set average at every data scale and model size

| Method | 0.5B · 100K | 0.5B · 400K | 0.5B · 800K | 7B · 800K |
|---|---|---|---|---|
| Equal | 27.03 | 30.51 | 31.78 | 60.02 |
| Grid | 27.98 | 26.13 | 31.17 | 59.46 |
| RegMix | 27.28 | 26.82 | 26.40 | 60.14 |
| DoReMi | 29.74 | 30.29 | 31.38 | 59.47 |
| ODM | 27.70 | 26.91 | 27.37 | 58.00 |
| DMO | 29.17 | 32.63 | 32.04 | 60.35 |
| **CausalMix-A** | **29.91** | **33.41** | **33.94** | 61.84 |
| **CausalMix-S** | 27.85 | 30.71 | 32.81 | **62.28** |

Average over six capability groups on development benchmarks; bold red marks the best value per column.

**+1.90**

CausalMix-A over the best baseline at 0.5B · 800K

**+1.93**

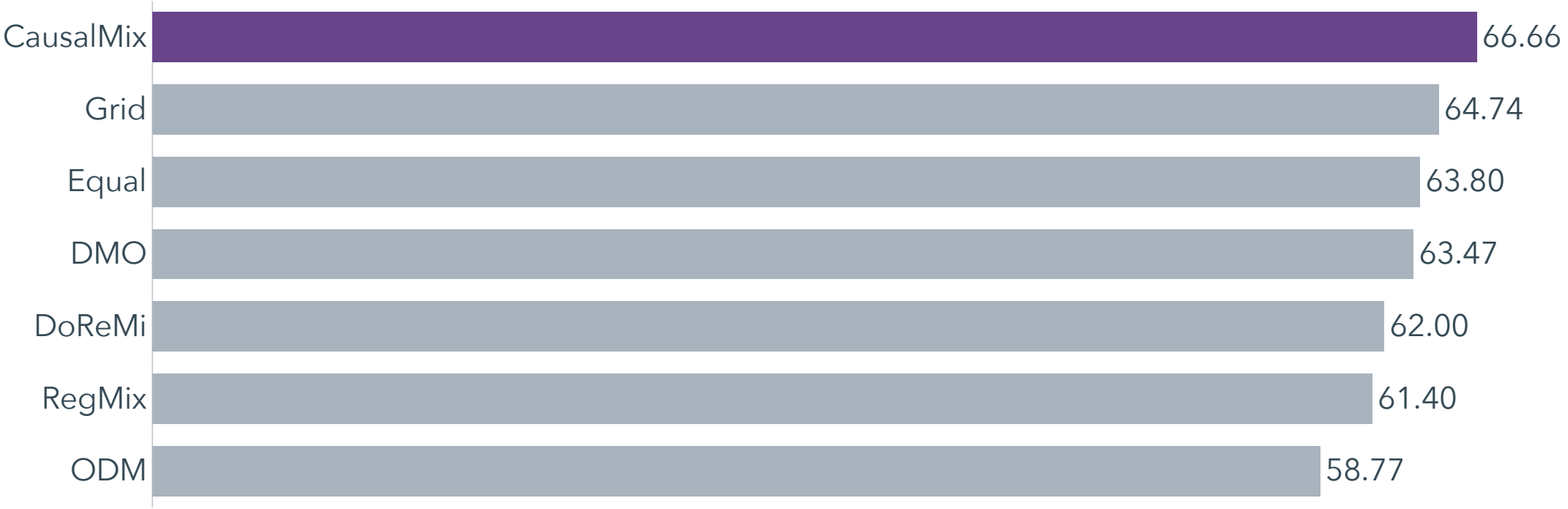
CausalMix-S over the best baseline at 7B · 800K

On unseen benchmarks, CausalMix stays within 1.5 points of the best baseline.

# Transfer to Long-CoT Data

No new proxy runs for a new data pool and model family

Fit on tulu-3 proxy runs, applied to unseen AM-Thinking-v1-Distilled math & code data; Qwen3-4B trained on 20K examples.




Average of GSM8K, MATH, HumanEval and MBPP; bars start at zero.

# Ablation and Covariates

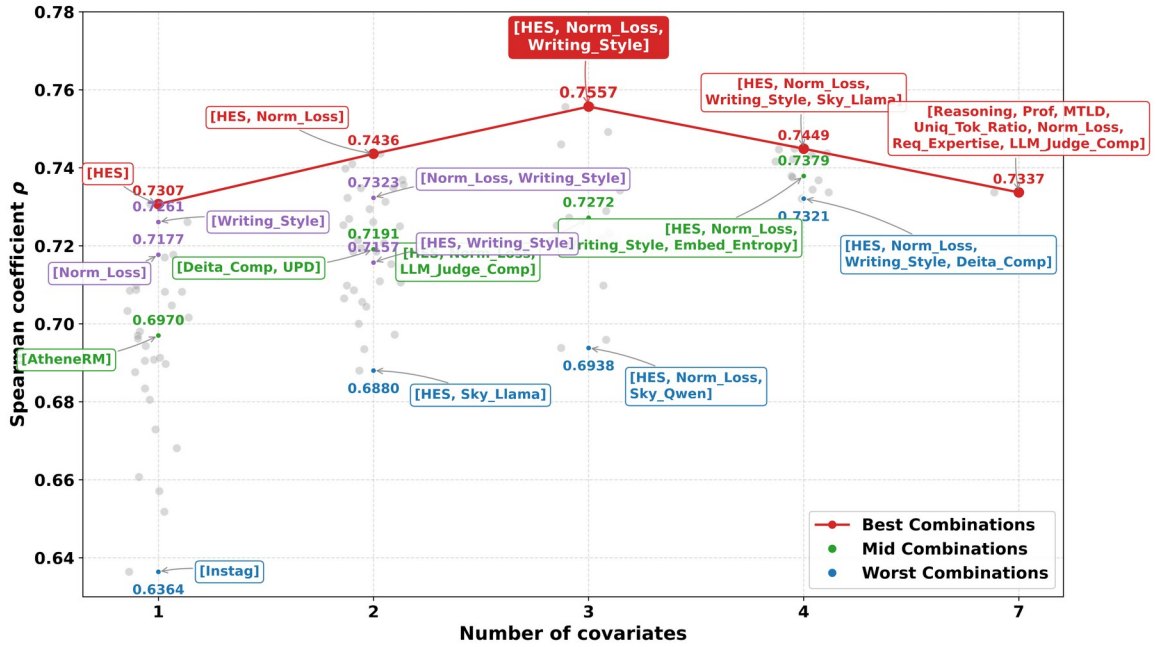
Data state, orthogonalization and covariate choice all matter

## Ablation

| Variant | 0.5B · 800K | 7B · 800K |
|---|---|---|
| w/o $X$ | 33.29 | 61.30 |
| w/o Orth. | 32.66 | 59.65 |
| **CausalMix-A** | **33.94** | 61.84 |
| **CausalMix-S** | 32.81 | **62.28** |

Development average; bold red marks the best. w/o $X$: a global map $Y = g(T)$ without the data state. w/o Orth.: direct regression $Y = f(X, T)$ without DML.

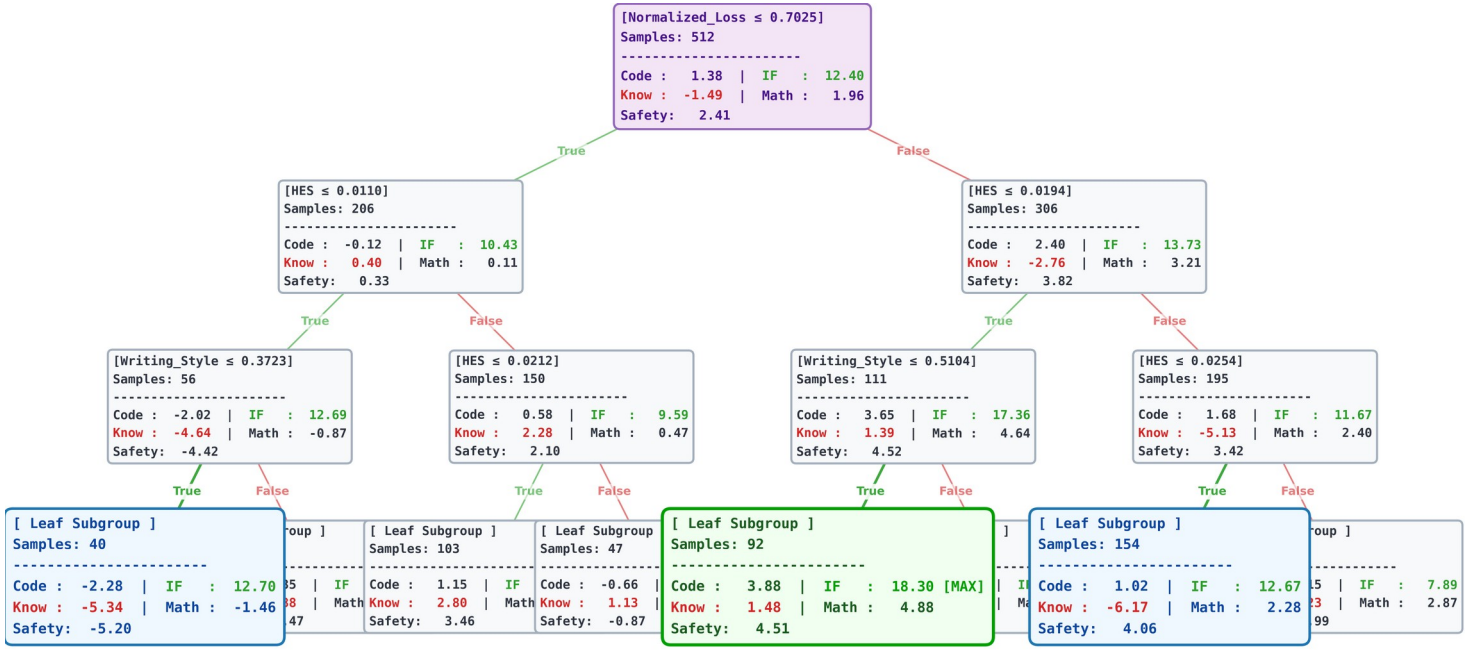
## Impact of covariates




HES + Normalized_Loss + Writing_Style (complexity, difficulty, quality) peak at Spearman $\rho = 0.756$ on 64 held-out runs.

# CATE Interpreter

What the learned mixing policy says




**IF data drives alignment:** positive return in every subgroup, up to 18.30

**Skill conflict:** Knowledge hurts on hard pools with high Normalized_Loss and HES (−6.17)

**Quality threshold:** Math, Code and Safety hurt on low-quality pools but gain at moderate Writing_Style and HES

# Takeaways

A causal, interpretable and transferable recipe for data mixing

1. Treat data mixing as a causal question: condition on the data state and estimate each domain's marginal return $\theta(X)$.

2. Double Machine Learning with a causal forest separates mixture effects from data-pool confounding.

3. 512 proxy runs on a 0.5B model set mixtures for an 800K pool, a 7B model and long-CoT data.

**Project page**

zinantang.pages.dev/causalmix