Can Released LLM Vocabularies Support Token-Level Estimation of Hidden Corpora?

Qingjie Zhang[1,2], Xingzhang Ren[2*], Zixuan Chen[1], Jinfeng Li[3], Yuefeng Chen[3], Yitong Yang[3], Hui Xue[3], Dayiheng Liu[2*], and Han Qiu[1*]

[1]Tsinghua University [2]Qwen Team, Alibaba Group [3]Alibaba Group

Emails: {qj-zhang24@mails., qiuhan@}tsinghua.edu.cn [*]Corresponding authors

Abstract

Pretraining corpus composition shapes LLM capabilities, but it often remains hidden even when model weights are released. Prior work has inferred corpus mixtures or traced specific token groups from released tokenizer vocabularies; in contrast, we estimate corpus ratios for arbitrary target tokens. We first show that BPE tokenizers trained on different corpora share stable token ID–ratio distributions, motivating distribution transfer from known corpora to a target tokenizer trained on hidden corpora. We then propose Quantile-Guided Density Estimation (QGDE), which approximates this distribution with multiple quantile trends and uses local density weighting to produce token-level estimates. In controlled settings and a realistic setting using the released SmolLM tokenizer, QGDE achieves mean relative errors as low as 3.00% for token-level estimation and 3.08% after aggregation into category-level mixtures. These results suggest that released tokenizer vocabularies provide a useful signal for fine-grained corpus estimation beyond coarse composition inference. The code of QGDE is available at https://github.com/qingjiesjtu/QGDE.

1 Introduction

The composition of pretraining corpora is essential for interpreting LLM performance, since it shapes basic capabilities such as multilingual and domain-specific performance (Xie et al., 2023; Hoffmann et al., 2022; Li et al., 2024; Petty et al., 2024). However, even models with released weights often leave the corpus opaque (Touvron et al., 2023; Bi et al., 2024; Liu et al., 2024). A more accessible signal is the tokenizer vocabulary, which is often released and commonly trained with byte-pair encoding (BPE) (Sennrich et al., 2016) on corpus statistics, as in the ChatGPT (Singh et al., 2025), Qwen (Bai et al., 2023; Yang et al., 2025), DeepSeek (Bi et al., 2024; Liu et al., 2024) families.

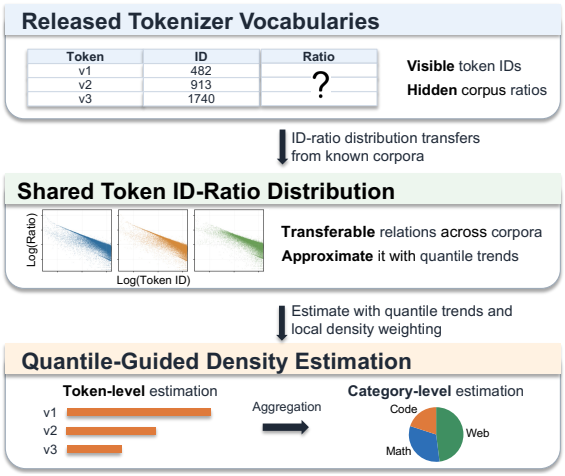

Figure 1: Overview of our work. QGDE transfers ID-ratio structure from known corpora to estimate token and category ratios of hidden corpora.

Prior work has shown that released BPE vocabularies can reveal corpus information[1]. Hayase et al. (2024) recover corpus mixture from BPE merge rules, but it remains coarse-grained rather than estimating the ratio of each token. Zhang et al. (2025) use token IDs to speculate about the prevalence of polluted Chinese tokens, but their setting targets a specific token group rather than general tokens. This motivates our central question:

Can released tokenizer vocabularies support general token-level estimation of hidden corpora?

Figure 1 illustrates our work. To estimate token ratios, we first compare BPE vocabularies trained on different corpora and find that their token ID–ratio distributions share a stable global shape. This motivates transferring this distribution from known corpora to a target tokenizer trained on hidden corpora. We then propose *Quantile-Guided Density Estimation* (QGDE), which fits multiple quantile trends to approximate the known ID–ratio distribution and enables token-level estimates for the target tokenizer. Our main contributions are:

[1]Tokenizer-training corpora may differ from pretraining corpora, but still reflect composition choices of developers.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 21582–21596
October 24-29, 2026 ©2026 Association for Computational Linguistics

- We show that token ID–ratio distributions are transferable across BPE vocabularies trained on different languages and domains, providing a usable signal for estimating hidden corpus ratios from released vocabularies.
- We propose QGDE, a general token-level estimator that fits multiple quantile trends to approximate the transferable distribution and uses local density weighting to predict corpus ratios for arbitrary target tokens.
- In both controlled settings and a realistic setting, QGDE achieves relative errors as low as 3.00% for token-level estimation and 3.08% for category-level estimation after aggregation, surpassing three compared methods.

2 Background

2.1 Related Work

Training data transparency and tokenizer signals. LLMs are trained on corpora whose exact composition is often undisclosed, making it difficult to interpret the data sources, language coverage, and domain coverage behind released models (Bommasani et al., 2023; Longpre et al., 2024; Dodge et al., 2021; Xu et al., 2024). Existing transparency methods often rely on model outputs or query access to reveal memorized examples, extract training data, or detect benchmark overlap (Carlini et al., 2021, 2022; Yang et al., 2023). In contrast, released tokenizers provide a corpus signal because they are learned from corpus statistics: frequent adjacent token pairs are merged earlier, and the resulting vocabulary and token IDs can encode distributional traces of the tokenizer-training corpus (Sennrich et al., 2016; Hayase et al., 2024; Zhang et al., 2025).

From coarse mixture to token-level estimation. Data Mixture Inference (DMI) (Hayase et al., 2024) treats released BPE merge rules as a composition signal: it compares tokenizer merge statistics with candidate corpora and estimates a small vector of category proportions. This output is inherently coarse-grained, over languages, domains, or data sources, rather than a ratio estimate for each token. PoCTrace (Zhang et al., 2025) exploits the relation between token IDs and frequency: it fits a single ID–ratio median trend to estimate the prevalence of polluted Chinese tokens. However, its use case remains tied to a specific token group and a coarse trend estimate. We instead study general token-level corpus ratio estimation for arbitrary tokens.

2.2 Problem Formulation

Let $\mathcal{V}^* = \{(v_i, t_i)\}_{i=1}^N$ denote the released vocabulary of a target tokenizer, where v_i is a token string and t_i is its token ID. The tokenizer-training corpus $\mathcal{C}^*$ is hidden. Let r_i^* denote the corpus ratio of v_i in $\mathcal{C}^*$. The hidden corpus therefore induces an unknown ID-to-ratio mapping $f^*(t_i) = r_i^*$.

We assume access to one or more *known corpora* $\mathcal{D}$. For each known corpus or mixture of known corpora, token ratios are observable after training and applying a BPE tokenizer (Sennrich et al., 2016; Hayase et al., 2024). These known corpora provide empirical ID–ratio pairs that can be used to approximate the unknown mapping f^*.

Our goal is to learn an estimator $\widehat{f}$ from the known ID–ratio pairs, such that for arbitrary tokens in the released target vocabulary,

$$\widehat{f}(t_i) = \widehat{r}_i \approx f^*(t_i) = r_i^*. \tag{1}$$

After estimating token-level ratios, we aggregate them over any token set $S \subseteq \mathcal{V}^*$, e.g., $\widehat{R}(S) = \sum_{i \in S} \widehat{r}_i$. When token sets are defined by languages or domains, this yields corpus mixture estimates.

3 Is ID-Ratio Relationship Transferable?

The formulation above relies on a basic assumption: ID–ratio pairs observed in known corpora should provide useful evidence for the hidden target corpus. Since BPE token IDs reflect merge order and thus corpus statistics (Sennrich et al., 2016; Hayase et al., 2024; Zhang et al., 2025), the key question is ***whether the ID–ratio relationship is transferable across tokenizers trained on different corpora***.

We compare ID–ratio distributions under two controlled settings: ***same domain but different languages, and same language but different domains***. Specifically, we train BPE tokenizers on English, French, Japanese, and Chinese slices from mC4 (Xue et al., 2021) for the language setting, and on FineWeb (Web) (Penedo et al., 2024), Wikipedia (Wiki) (Wikimedia Foundation, 2023), CodeParrot (Code) (Hugging Face, 2021), and OpenWebMath (Math) (Paster et al., 2024) for the domain setting. For each tokenizer, we remove the initial vocabulary tokens, since special tokens and alphabet symbols do not reflect BPE merge order. We then compute each token's ratio as its count divided by the total token count.

We plot the token ID–ratio distributions in log–log space in Figure 2. The scatter plots show that

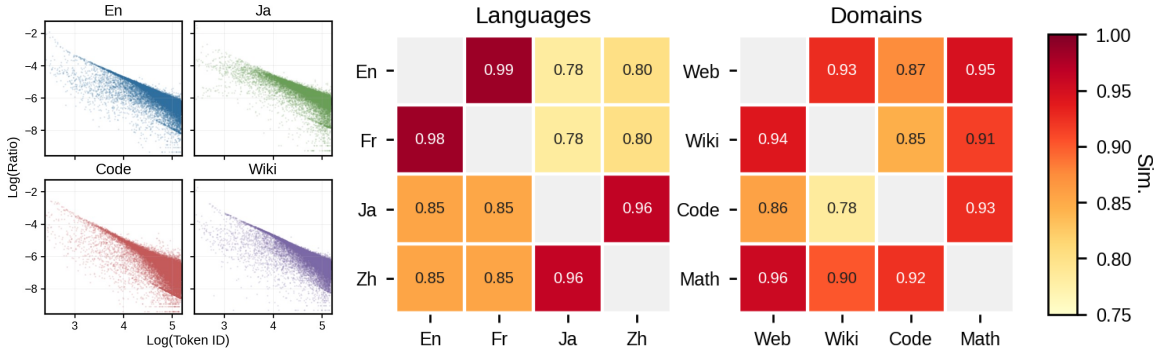

Figure 2: Token ID–ratio distributions are transferable across different corpora. **Left**: ID–ratio scatter plots in log–log space. **Middle**: Similarity across different languages. **Right**: Similarity across different domain.

the distributions are visually similar, even for English and Japanese, and for Code and Wiki, which differ substantially in language or domain structure (see Figure 7 for all eight corpora). This suggests a transferable ID–ratio distribution.

To quantify this similarity, we convert each scatter plot into a two-dimensional probability density over $(\log \mathrm{ID}, \log \mathrm{ratio})$ (see Appendix B) and define a directional transfer similarity score. Let P_S and P_T denote the ID–ratio densities of a source tokenizer S and a target tokenizer T. We compute

$$\mathrm{Sim}(S \to T) = \exp\left(-\frac{D_{\mathrm{KL}}(P_T \| P_S)}{H(P_T)}\right), \quad (2)$$

where $H(P_T)$ is the entropy of the target density and D_{KL} denotes Kullback–Leibler divergence (Murphy, 2022). This score normalizes the transfer divergence by the target entropy and maps it to $(0, 1]$, where larger values indicate that the source distribution better explains the target distribution.

The heatmap in Figure 2 shows that ***token ID–ratio distributions are broadly transferable***. In the language setting, transfer is strongest between languages with more similar linguistic structures: English and French are nearly interchangeable, and Japanese and Chinese are also highly similar. Transfer across these groups is weaker, but still substantial, with similarities around 0.8 in the harder English/French-to-Japanese/Chinese directions. In the domain setting, all pairs remain highly similar, with the largest gap appearing between Code and Wiki, mirroring the larger structural gap between programming language and encyclopedic text. Additional single- and mixed-source transfer profiles are reported in Appendix C, and transfer across tokenizer families is examined in Appendix I. Overall, the ID–ratio relationship has a stable global shape, motivating an estimator that exploits the shared global ID–ratio trend across corpora.

4 Quantile-Guided Density Estimation

Since token ID–ratio distributions share a global shape, we estimate token ratios by transferring this distributional structure from known corpora with observed token ratios. Such transfer is inevitably imperfect: ***the key is to preserve the shared ID–ratio distribution***.

Inspired by prior work (Zhang et al., 2025) that uses quantile regression to obtain ratio ranges from a single median or boundary curve, we fit multiple quantile trends to better approximate the shared ID–ratio distribution. We then use local density weighting to convert these quantile trends into a token point estimate, as illustrated in Figure 3.

4.1 Fitting ID–Ratio Trends with Quantiles

Following Zipf's law (Piantadosi, 2014; Saichev et al., 2009), which states that word frequency is approximately inversely proportional to frequency rank, a log–log transformation turns this inverse relationship into an approximately linear trend. We therefore model token ID and corpus ratio in log–log space, where quantile regression can fit linear trends at different quantile levels.

Given a tokenizer trained on known corpora with observed token counts, we represent each token j as $(x_j, y_j) = (\log t_j, \log r_j)$, where t_j is the token ID and r_j is the token's corpus ratio. A single median curve, corresponding to the 0.5 quantile, can capture the central tendency of this relation. However, *it reduces tokens with similar IDs to a single typical ratio, discarding their local ratio variation. This local variation is exactly the distributional signal needed for token-level estimation.*

We instead model the shared ID–ratio relationship ***with a family of quantile trends***. Let $\mathcal{T}$ denote a set of candidate quantile levels. For each $\tau \in \mathcal{T}$, we fit a log-linear quantile curve over the known ID–ratio points using standard quantile regression

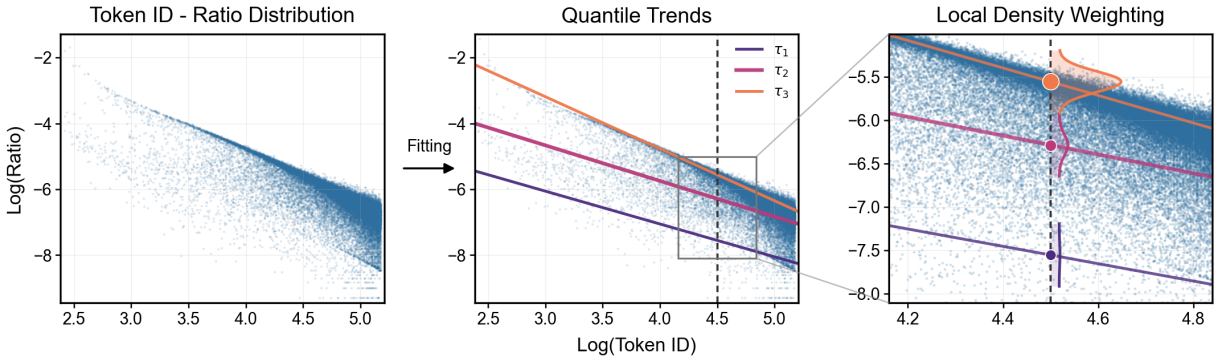

Figure 3: Overview of quantile-guided density estimation. **Left**: the ID–ratio distribution in log–log space. **Middle**: multiple quantile trends approximate the global ID–ratio distribution and produce candidate estimates at a target token ID. **Right**: Gaussian density kernels around each candidate assign local weights to the estimates.

(Regression, 2017), $q_\tau(x) = a_\tau + b_\tau x$ (see details in Appendix A). For a target token with ID t_i, the τ-th trend gives a candidate log-ratio estimate:

$$z_{i,\tau} = q_\tau(\log t_i) = a_\tau + b_\tau \log t_i. \tag{3}$$

The quantile family generalizes single-curve token-ID estimators: lower quantiles describe conservative low-ratio hypotheses, upper quantiles describe high-ratio hypotheses, and intermediate quantiles describe the dense central region. Thus, at each token ID, a dense quantile family defines multiple plausible candidate estimates. The remaining question is which quantile levels best approximate the shared ID–ratio distribution.

4.2 Selecting Quantile Anchors

To avoid redundant or poorly supported trends, we select a small set of representative quantile anchors $\mathcal{T}_K^\star = \{\tau_1, \ldots, \tau_K\}$. *These anchors should cover the global ID–ratio distribution without collapsing onto redundant regions or passing through consistently sparse regions.* We therefore select anchors by directly maximizing their coverage over the known ID–ratio points $\mathcal{P} = \{(x_j, y_j)\}_{j=1}^n$.

For a candidate anchor set $\mathcal{T}_K$, a point is covered if it falls within a vertical band of width h_y around at least one selected quantile trend. We define Quantile Anchor Coverage C as

$$C(\mathcal{T}_K) = \sum_{(x_j,y_j)\in\mathcal{P}} \mathbf{1}\left[\min_{\tau\in\mathcal{T}_K} |y_j - q_\tau(x_j)| < h_y\right]. \tag{4}$$

This score counts each known ID–ratio point at most once, even if it is close to multiple selected trends, so redundant anchors receive little extra benefit. We choose the anchor set by maximizing this Quantile Anchor Coverage:

$$\mathcal{T}_K^\star = \arg\max_{\mathcal{T}_K\subset\mathcal{T},\,|\mathcal{T}_K|=K} C(\mathcal{T}_K). \tag{5}$$

In implementation, we perform a grid search over quantile levels. Because the maximization depends on both the selected quantile set and the number of anchors K, Section 5 analyzes which anchors are selected and how many anchors are needed.

4.3 Local Density Weighting

After selecting quantile anchors, each target token has multiple candidate estimates $\{z_{i,\tau}\}_{\tau\in\mathcal{T}_K^\star}$. These candidates are plausible under the global ID–ratio trends, but the global trends do not determine how much each candidate should contribute to a particular token. Therefore, *we assign them soft weights by measuring how much local density support each candidate receives from nearby known ID–ratio points*.

For target token t_i, we collect nearby known ID–ratio points $\mathcal{N}_i = \{(x_j, y_j) \mid |x_j - \log t_i| < h_x\}$. Within this local ID window, we compute the unnormalized local support using Gaussian kernel (Silverman, 2018):

$$W_{i,\tau} = \sum_{(x_j,y_j)\in\mathcal{N}_i} \exp\left(-\frac{(y_j - z_{i,\tau})^2}{2h_y^2}\right). \tag{6}$$

This kernel is a soft version of the vertical neighborhood used in anchor coverage, centered at the candidate estimates $z_{i,\tau}$. This weighting matches the intuition in Figure 3: at a fixed token ID, Gaussian density kernels compare how strongly the surrounding points support each quantile candidate.

By normalizing the weights across all quantile trends, the estimated ratio of token t_i is the density-weighted average of the candidate estimates:

$$\widehat{y}_i = \sum_{\tau\in\mathcal{T}_K^\star} \frac{W_{i,\tau}}{\sum_{\tau'\in\mathcal{T}_K^\star} W_{i,\tau'}} z_{i,\tau}. \tag{7}$$

Thus, the global quantile trends provide candidate estimates, and local density weighting combines

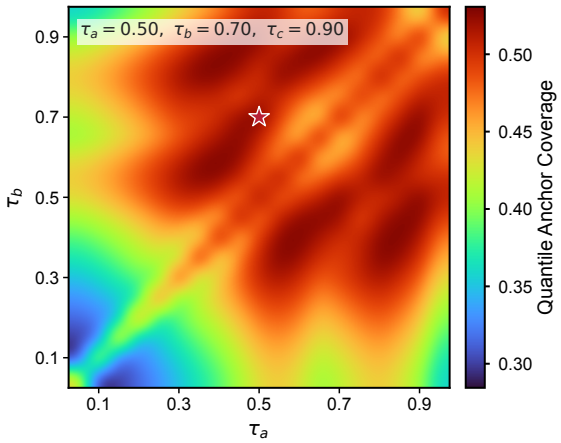

Figure 4: Pairwise projection of Quantile Anchor Coverage for selecting three quantile anchors. The star marks the selected triplet $(0.50, 0.70, 0.90)$.

them using token-specific neighborhood evidence. As a result, it turns the range-style signal used in prior token-ID frequency estimation (Zhang et al., 2025) into a fine-grained token-level point estimate.

5 Quantile Anchor Configuration

In Equation 5, we select quantile anchors to cover the known ID–ratio distribution while avoiding redundant or poorly supported trends. Before evaluating token-level estimates, we first examine how this Quantile Anchor Coverage (QAC) objective configures the anchors in practice. The following subsections address two practical questions.

5.1 Which Anchors Cover Better?

We first inspect the fixed-K selection problem. Taking $K = 3$ as an example, we grid search over candidate triplets using the QAC objective in Equation 5. Since the triplet search space is three-dimensional, Figure 4 visualizes a pairwise projection: each cell fixes two anchors (τ_a, τ_b) and reports the best coverage obtained by any triplet containing that pair.

Figure 4 shows that coverage is substantially lower when anchors concentrate in sparse or redundant regions. By contrast, the high-coverage area lies around middle-to-high quantile levels. In this $K = 3$ configuration, the triplet $(0.50, 0.70, 0.90)$ achieves the highest coverage, covering 53.1% of the sampled known ID–ratio points under the chosen vertical bandwidth. This indicates that QAC does not simply spread anchors uniformly across quantile levels; instead, it favors anchors whose trends jointly pass through well-supported regions of the known ID–ratio distribution.

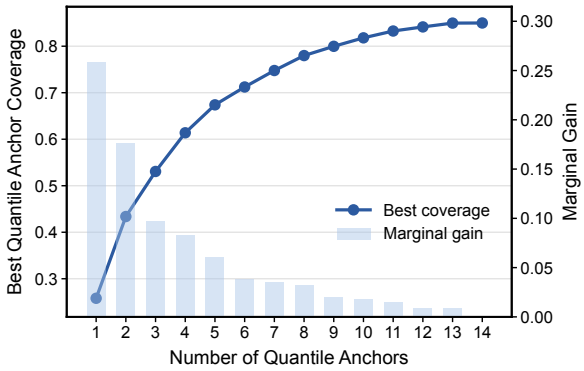

Figure 5: Effect of the number of quantile anchors on Quantile Anchor Coverage. Coverage increases with K, but marginal gains quickly saturate.

5.2 How Many Anchors Are Enough?

We then study how the number of quantile anchors affects coverage. For each anchor number K, we maximize $C(\mathcal{T}_K)$ and report both the best coverage and the marginal gain over $K - 1$ anchors in Figure 5. Coverage increases as more anchors are added, but the marginal gain decreases rapidly and becomes nearly zero at $K = 14$. This saturation suggests that additional anchors eventually pass through regions that are already covered by the selected trends, rather than adding substantial new support from the known ID–ratio distribution.

Overall, QAC favors complementary anchors rather than sparse or redundant ones. Its coverage then saturates as K grows, indicating diminishing returns from adding more anchors. A target-independent held-out procedure for selecting K is reported in Appendix F.

6 Token-Level Ratio Estimation

We now test whether QGDE turns the quantile trends into accurate token-level ratio estimates.

6.1 Evaluation Setting

The evaluation is under different source compositions. Here, source refers to the known corpus mixture used to fit the estimator, while the target corpora are held fixed as uniform mixtures and used only for evaluation.

We use the same language and domain categories as in Section 3, but evaluate on target corpora drawn from different datasets. For languages, mC4 serves as the source side and OSCAR (Suarez et al., 2020) as the target side over English, French, Japanese, and Chinese. For domains, Web, Wiki, Code, and Math are paired with target corpora from the same broad domains: RedPajama-

Source	Single-source					70%-mixed source					Target-like
	S_1-only	S_2-only	S_3-only	S_4-only	Avg.±Std.	S_1-major	S_2-major	S_3-major	S_4-major	Avg.±Std.	Uniform
Language Sources (S_1 =En, S_2 =Fr, S_3 =Zh, S_4 =Ja)											
Transfer	21.43	21.40	31.25	37.71	27.95±6.92	22.19	19.58	33.53	22.33	24.41±5.38	21.72
PoCTrace	24.94	25.01	10.33	9.40	17.42±7.56	18.82	14.21	8.27	9.75	12.76±4.12	10.45
QGDE Avg.	**5.80**	**5.84**	**6.33**	**5.21**	**5.80±0.40**	**5.93**	**3.91**	**4.65**	**4.64**	**4.78±0.73**	**4.68**
$K = 3$	14.11	18.05	9.17	10.95	13.07±3.38	17.54	10.02	7.29	10.78	11.41±3.77	11.42
$K = 4$	14.19	11.95	7.98	7.24	10.34±2.85	9.65	5.79	6.57	6.76	7.19±1.46	7.29
$K = 5$	7.46	7.91	7.99	6.08	7.36±0.76	7.55	3.47	5.90	5.68	5.65±1.45	5.75
$K = 6$	4.48	4.44	7.08	4.65	5.16±1.11	8.36	4.61	4.55	4.18	5.43±1.70	3.94
$K = 7$	3.46	3.44	6.27	5.03	4.55±1.19	4.24	3.26	4.90	4.47	4.22±0.60	4.44
$K = 8$	4.69	4.20	5.49	4.95	4.83±0.46	3.83	2.92	4.03	3.40	3.54±0.43	3.45
$K = 9$	3.95	3.76	5.63	4.03	4.34±0.75	3.03	2.79	3.89	3.50	3.31±0.42	3.39
$K = 10$	3.85	3.14	4.84	4.07	3.98±0.61	3.53	2.79	3.39	3.52	3.31±0.31	3.51
$K = 11$	3.20	3.17	5.10	3.54	3.75±0.79	3.57	2.82	3.51	3.09	3.25±0.31	3.00
$K = 12$	2.91	2.94	5.29	3.76	3.72±0.97	3.09	2.77	3.71	3.22	3.20±0.34	3.09
$K = 13$	3.54	3.43	5.49	3.98	4.11±0.82	3.26	2.76	3.90	3.44	3.34±0.41	3.28
$K = 14$	3.78	3.66	5.64	4.23	4.33±0.79	3.50	2.93	4.09	3.67	3.55±0.42	3.55
Domain Sources (S_1 =Web, S_2 =Wiki, S_3 =Math, S_4 =Code)											
Transfer	13.76	13.75	53.02	12.78	23.33±17.15	10.42	17.59	20.96	9.97	14.73±4.70	14.19
PoCTrace	30.41	27.50	37.72	21.13	29.19±5.96	27.51	26.34	25.22	22.20	25.32±1.97	24.32
QGDE Avg.	**11.45**	**8.06**	**16.77**	**24.28**	**15.14±6.12**	**7.34**	**6.31**	**8.71**	**11.28**	**8.41±1.86**	**7.44**
$K = 3$	29.08	18.66	28.44	31.44	26.91±4.89	22.21	17.38	24.22	23.79	21.90±2.71	22.24
$K = 4$	25.12	17.29	27.43	32.78	25.66±5.57	19.59	10.85	21.70	22.15	18.57±4.56	11.89
$K = 5$	17.87	10.45	26.06	31.42	21.45±7.97	4.66	4.89	14.52	17.99	10.52±5.87	12.76
$K = 6$	18.60	11.22	23.76	30.94	21.13±7.21	5.91	4.85	7.64	15.04	8.36±3.98	5.42
$K = 7$	13.19	7.03	20.64	31.67	18.13±9.18	4.47	4.48	4.73	10.96	6.16±2.78	4.44
$K = 8$	5.33	4.52	20.43	31.81	15.52±11.34	4.52	4.69	4.67	11.45	6.33±2.96	4.44
$K = 9$	5.15	4.50	16.77	31.28	14.43±10.89	4.48	4.89	4.47	11.02	6.21±2.78	4.52
$K = 10$	4.49	4.66	8.63	30.05	11.96±10.58	4.45	4.84	4.45	5.00	4.69±0.24	4.76
$K = 11$	4.70	4.59	8.81	10.63	7.18±2.62	4.49	5.01	4.58	4.49	4.64±0.22	4.99
$K = 12$	4.54	4.67	7.32	10.53	6.76±2.44	4.45	4.71	4.46	4.45	4.52±0.11	4.67
$K = 13$	4.63	4.57	7.61	9.54	6.59±2.10	4.45	4.60	4.51	4.51	4.52±0.05	4.60
$K = 14$	4.65	4.52	5.30	9.27	5.94±1.95	4.46	4.56	4.53	4.51	4.52±0.04	4.54

Table 1: Mean relative error (MRE) (%) of token-level ratio estimation across language and domain source mixtures. Single-source uses one known corpus; 70%-mixed uses a 70%-dominant known-corpus mixture, with the remaining three sources mixed equally; Target-like uses the same uniform mixture as the target corpora. Darker green indicates higher error in the QGDE K-ablation rows.

C4 (Weber et al., 2024), BookCorpus (Zhu et al., 2015), RedPajama-GitHub (Weber et al., 2024), and FineWebMath (Allal et al., 2025).

We report mean relative error in Table 1, where lower values indicate better estimates. We compare QGDE with two baselines: direct ID-ratio transfer, which copies the source ratio profile by token position, and PoCTrace (Zhang et al., 2025), which estimates ratios from a single median ID–ratio trend.

6.2 Token-Level Results

Table 1 yields four main observations.

QGDE outperforms baselines. QGDE consistently improves token-level estimation over both baselines. Direct ID-ratio transfer copies the source ratio profile and therefore remains high-error, showing that distributional transferability does not justify token-wise ratio copying. PoCTrace avoids direct copying by fitting a median ID–ratio trend, but a single trend cannot represent the vertical spread of plausible ratios at each token ID. In contrast, QGDE combines multiple quantile trends with local density weighting. The later K-rows are substantially lower than both baselines in most source-mixture settings, especially for mixed known-corpus settings.

More quantile anchors help, then saturate. The ablation on K confirms that multiple quantile anchors are necessary, but that their benefit saturates. Moving from $K = 3$ to larger K sharply reduces error, especially in the domain block: the single-source average decreases from 26.91 to 5.94, and the 70%-mixed average decreases from 21.90 to 4.52. Later rows fluctuate within a narrower band, so the best K is not universal across source compositions; nevertheless, $K = 14$ provides a reasonable high-coverage default once the QAC gain has saturated. This echoes the QAC analysis in Section 5: once additional anchors provide little new coverage of the known ID–ratio distribution, they also yield diminishing improvements in token-level estimation error.

Mixed sources help, but exact ratios matter less. Mixed sources are generally preferable to single-source settings because they expose the estimator to a broader ID–ratio distribution. This is most visible in the domain block, where the QGDE average drops from 15.14 for single-source settings to 8.41 for 70%-mixed settings. Within the mixed-source group, however, the exact dominant component matters much less: at $K = 14$, the 70%-mixed columns have low standard deviation, with Std. of 0.42 for languages and 0.04 for domains. Even matching the target mixture exactly (Target-like at the last column) is not always optimal; what matters more is using a mixed source that covers multiple source components.

Language ratios are easier to predict than domain ratios. The language block reaches low error with fewer anchors, whereas the domain block has much larger errors at small K and only approaches a similar range after more anchors are used. This difference reflects how closely the source and target ID–ratio relationships match. In the language setting, mC4 and OSCAR differ as corpora, but the language-specific signals that shape token IDs are largely stable across them. In the domain setting, the matched source and target corpora are less aligned. Source and target corpora may differ in collection pipelines, so the same broad domain label does not guarantee a similar ID–ratio relationship.

7 Aggregating Token Ratios into Mixtures

A useful token-level estimator should also support corpus mixture estimation. We test this by aggregating QGDE's token-level estimates into language or domain proportions.

7.1 Aggregation Procedure

We convert target token ratios into category proportions by distributing each estimated token ratio $\widehat{r}_i$ *according to how the token appears across known source categories*. For example, a token that appears mostly in one category contributes most of its estimated ratio to that category, while a token that appears across all categories is split according to its relative counts in the known corpora. Formally, we define the category assignment weight as

$$\pi_{c,i} = \frac{n_{c,i}}{\sum_{c' \in \mathcal{C}} n_{c',i}}, \qquad c \in \mathcal{C}. \tag{8}$$

Here, $\mathcal{C}$ is the set of source categories, and $n_{c,i}$ is the count of target token v_i in the known corpus for category c.

We then normalize the estimated token ratios over token set $\mathcal{I}$ and distribute each token ratio to categories using $\pi_{c,i}$. The estimated mixture proportion of category c is therefore

$$\widehat{\alpha}_c = \sum_{i \in \mathcal{I}} \frac{\widehat{r}_i}{\sum_{j \in \mathcal{I}} \widehat{r}_j} \pi_{c,i}, \qquad c \in \mathcal{C}. \tag{9}$$

7.2 Category-Level Results

Figure 6 evaluates category-level mixture estimation under uniform language and domain targets. We compare QGDE aggregation with DMI (Hayase et al., 2024), a source-independent baseline that specifically designed for estimating macro mixture proportions from tokenizer statistics.

Token-level estimates support mixture estimation. The figure shows that QGDE token-level ratios can be aggregated into meaningful category-level proportions. Across both language and domain targets, the QGDE pies track the ground-truth mixtures closely than the baseline estimates. This indicates that the token-level signal recovered by QGDE is not only useful for individual token prediction, but also remains informative after aggregation to categories.

QGDE outperforms baseline DMI. DMI gives visibly skewed estimates: in the language setting, it overestimates English and French while underestimating Chinese and Japanese; in the domain setting, it strongly overestimates Web and underestimates the remaining categories. QGDE substantially reduces these errors, lowering language error from 9.09 to about 3.0 and domain error from 15.14 to a much smaller range.

Anchor gains are weaker after aggregation. The effect of increasing K is less pronounced at the category level than at the token level. In the language setting, error decreases from $K = 3$ to $K = 8$, but changes only slightly at $K = 14$. In the domain setting, the trend is even less monotonic, because aggregation can shift estimated ratios among overlapping domain vocabularies even when token-level predictions improve. Thus, additional anchors still help by improving the underlying token estimates, but their gains are partially smoothed or redistributed by category-level aggregation. The full source-composition and K-sweep results are reported in Table 5.

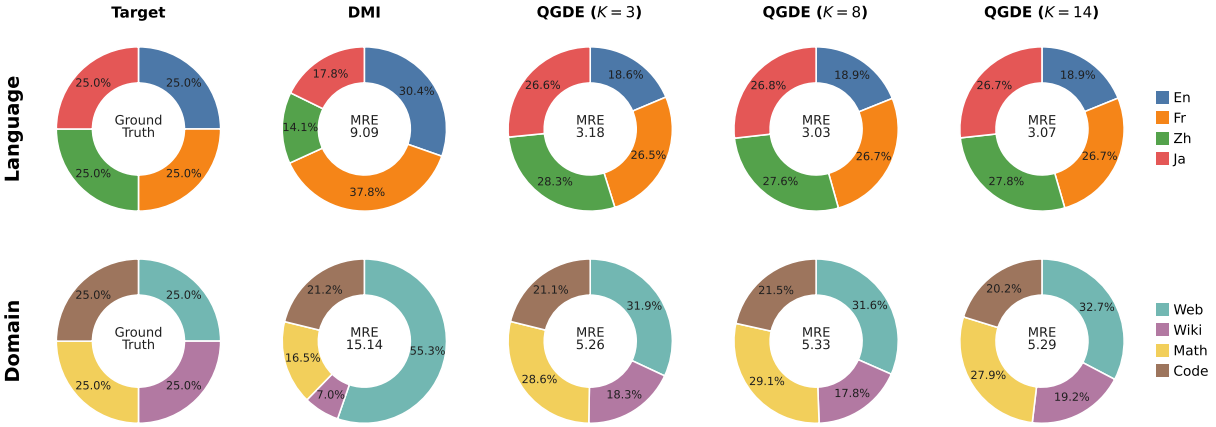

Figure 6: Category-level mixture estimation, where QGDE matches the ground truth more closely in both language and domain settings. Each setting compares the ground truth, DMI, and QGDE at three K values; center labels report mean relative error (MRE).

8 Validation on the SmolLM Tokenizer

Beyond the controlled language and domain settings, we also test QGDE in a realistic setting: *a released tokenizer along with training corpora*. Such validation is uncommon because LLM releases rarely include training corpora. SmolLM (Allal et al., 2025) is a useful exception.

We use the SmolLM tokenizer as the target tokenizer. Token-level ground truth is computed from the training corpus, and category-level ground truth uses the released component proportions: FineWeb-edu 87.30%, Cosmopedia-v2 11.11%, and Python-edu 1.59%. Since the exact component corpora cannot be used as known corpora, we fit the estimators on corpora matched to the three components: RedPajama-C4 (Weber et al., 2024), a Wikipedia–arXiv mixture (Wikimedia Foundation, 2023; Cornell University, 2020), and CodeParrot (Hugging Face, 2021), respectively. We evaluate three known-corpus mixture ratios over these proxies: a uniform mixture, a pretrain-like mixture based on (Soldaini et al., 2024), and a target-like mixture that matches the released SmolLM component proportions as a diagnostic setting. Table 2 shows the results.

At the token-level, QGDE outperforms both direct ID-ratio transfer and PoCTrace, reaching 5.72–5.78 MRE. The small spread across Uniform, Pretrain-like, and Target-like source mixture also echoes the controlled experiments: once the known-corpus mixture covers the relevant components, the exact mixture ratio is less important. An additional validation of Pythia (Biderman et al., 2023) is reported in Appendix H.

At the category level, QGDE also improves over the source-independent DMI baseline, reducing error from 9.11 to about 5.9–6.1. These estimates are less exact than in the controlled category-level experiments because the component proportions are highly imbalanced; in particular, Python-edu accounts for only 1.59%. Nevertheless, QGDE still recovers a better mixture estimate. A direct proxy-corpus replacement experiment is reported in Appendix G.

Source	Uniform	Pretrain-like	Target-like
Token-level			
Transfer	9.06	10.53	7.71
PoCTrace	13.84	18.68	20.33
QGDE	**5.78**	**5.73**	**5.72**
Category-level			
DMI	9.11	9.11	9.11
QGDE	**6.08**	**5.90**	**5.93**

Table 2: Validation on the released SmolLM tokenizer, where QGDE achieves the lowest error for both token-level and category-level estimation.

9 Conclusion

Released LLM vocabularies provide a useful signal for estimating hidden corpus composition beyond coarse category proportions. We show that BPE tokenizers share stable token ID–ratio distributions across corpora, and introduce QGDE to transfer this structure through quantile trends and local density weighting. Across controlled settings and the released SmolLM tokenizer, QGDE achieves relative errors as low as 3.00% for token-level estimation and 3.08% after aggregation into category-level mixtures.

Limitations

Scarcity of ground truth for released LLM tokenizers. ChatGPT, Qwen, and DeepSeek release tokenizer vocabularies, but not their training corpora. This prevents direct evaluation of token-level ratio estimates on these models. We therefore validate QGDE in controlled settings and on SmolLM (also Pythia in Appendix H), rare released tokenizers with available training data.

QGDE is designed for BPE tokenizers. As stated in Section 1, QGDE is designed for BPE tokenizers. The diagnostic in Appendix I shows weaker transfer to Unigram and WordPiece tokenizers, so applying QGDE unchanged to those constructions is outside the scope. Also, manually concatenated BPE tokenizers is out of scope. Nevertheless, this BPE scope still covers many mainstream released decoder-only LLM tokenizers.

Ethics Statement

ACL Ethics Policy is respected in this work. This work studies corpus ratio estimation from released tokenizer vocabularies and known corpora. We use publicly available or controlled corpora for research purposes, and we respect the terms, conditions, and copyright requirements of the corresponding data sources. No human subjects or private personal data are involved. The proposed methods are intended for research use, transparency analysis, and auditing of corpus composition signals from released tokenizers.

We adhere to the Association for Computational Linguistics (ACL) guidelines on responsible NLP research[2], with particular attention to transparency, research-use framing, and responsible handling of corpus-derived evidence.

Use of AI Assistants

The authors used AI assistants for language polishing, LaTeX editing, and phrasing suggestions during paper preparation. All substantive claims, experimental results, analyses, citations, and final text were reviewed and verified by the authors.

Acknowledgements

This work was supported by Alibaba Group through Alibaba Innovative Research Program.

[2] https://aclrollingreview.org/responsibleNLPresearch/

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. 2025. Smollm2: When smol goes big–data-centric training of a small language model.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pages 2397–2430. PMLR.

Rishi Bommasani, Kevin Klyman, Shayne Longpre, Sayash Kapoor, Nestor Maslej, Betty Xiong, Daniel Zhang, and Percy Liang. 2023. The foundation model transparency index. *arXiv preprint arXiv:2310.12941*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650.

Cornell University. 2020. arXiv dataset. https://www.kaggle.com/datasets/Cornell-University/arxiv.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Jonathan Hayase, Alisa Liu, Yejin Choi, Sewoong Oh, and Noah A Smith. 2024. Data mixture inference

attack: Bpe tokenizers reveal training data compositions. *Advances in Neural Information Processing Systems*, 37:8956–8983.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Hugging Face. 2021. CodeParrot. https://huggingface.co/codeparrot.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. 2024. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Shayne Longpre, Robert Mahari, Anthony Chen, Naana Obeng-Marnu, Damien Sileo, William Brannon, Niklas Muennighoff, Nathan Khazam, Jad Kabbara, Kartik Perisetla, et al. 2024. A large-scale audit of dataset licensing and attribution in ai. *Nature Machine Intelligence*, 6(8):975–987.

Kevin P. Murphy. 2022. *Probabilistic Machine Learning: An Introduction*. MIT Press.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2024. Openwebmath: An open dataset of high-quality mathematical web text.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale.

Jackson Petty, Sjoerd van Steenkiste, and Tal Linzen. 2024. How does code pretraining affect language model task performance? *arXiv preprint arXiv:2409.04556*.

Steven T Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130.

Quantile Regression. 2017. Handbook of quantile regression. *Boca Raton, FL, USA: CRC*.

Alexander I Saichev, Yannick Malevergne, and Didier Sornette. 2009. *Theory of Zipf's law and beyond*, volume 632. Springer Science & Business Media.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1715–1725.

Bernard W Silverman. 2018. *Density estimation for statistics and data analysis*. Routledge.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast wordpiece tokenization. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 2089–2103.

Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1703–1714.

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 arXiv:2307.09288*.

Maurice Weber, Daniel Y Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. 2024. Redpajama: an open dataset for training large language models.

Wikimedia Foundation. 2023. Wikipedia dumps. https://dumps.wikimedia.org/.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. volume 36, pages 69798–69818.

Cheng Xu, Shuhao Guan, Derek Greene, M Kechadi, et al. 2024. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 483–498.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*.

Qingjie Zhang, Di Wang, Haoting Qian, Liu Yan, Tianwei Zhang, Ke Xu, Qi Li, Minlie Huang, Hewu Li, and Han Qiu. 2025. Speculating llms' chinese training data pollution from their tokens. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26124–26144.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.

A Quantile Trend Fitting Details

In Section 4.1, QGDE fits one log-linear trend for each quantile level over the known ID–ratio points using quantile regression (Regression, 2017). This section gives the optimization objective used to fit those trends.

Let $\mathcal{P} = \{(x_j, y_j)\}_{j=1}^{n}$ denote the known ID–ratio points, where $x_j = \log t_j$, $y_j = \log r_j$, t_j is the token ID, and r_j is the observed ratio in the known corpus. For each quantile level $\tau \in \mathcal{T}$, we estimate the trend coefficients by

$$(a_\tau, b_\tau) = \arg\min_{a,b} \sum_{j=1}^{n} \rho_\tau \left(y_j - a - bx_j\right), \quad (10)$$

where $\rho_\tau(\cdot)$ is the asymmetric loss:

$$\rho_\tau(u) = \begin{cases} \tau u, & u \geq 0, \\ (\tau - 1)u, & u < 0. \end{cases} \quad (11)$$

The fitted trend is $q_\tau(x) = a_\tau + b_\tau x$. The asymmetric loss encourages approximately a τ fraction of known points to lie below the trend, so different τ values trace different vertical levels of the ID–ratio distribution. A single median trend captures only the central tendency; QGDE fits a family of trends so that the vertical spread of plausible ratios is preserved before local density weighting.

B Transfer Similarity Computation

In Section 3, Figure 2 quantifies whether ID–ratio distributions transfer across tokenizers. We compute this score by converting each scatter plot into a probability density P_D over ID–ratio space.

For each tokenizer D, each token v_i is represented as $(x_i, y_i) = (\log t_i, \log r_i)$, where t_i is its token ID and r_i is its corpus ratio. We place all tokenizers on a shared grid $\mathcal{B}$ and convert each tokenizer into a smoothed two-dimensional histogram:

$$P_D(b) = \frac{\sum_{v_i \in V_D} \mathbf{1}[(x_i, y_i) \in b] + \epsilon}{|V_D| + \epsilon|\mathcal{B}|}, \qquad b \in \mathcal{B}. \quad (12)$$

Here, ϵ is a small smoothing constant. Setting $D = S$ or $D = T$ gives the source and target densities P_S and P_T used in the main text.

The score is directional: $\mathrm{Sim}(S \to T)$ asks how well the source density P_S explains the target density P_T. We therefore compute $D_{\mathrm{KL}}(P_T \| P_S)$, normalize it by the target entropy $H(P_T)$, and map the result to $(0, 1]$, where higher values indicate stronger transfer similarity (Murphy, 2022).

Source Ratio	Target			
Web:Wiki:Code:Math	Web	Wiki	Code	Math
100:0:0:0	1.00	0.93	0.87	0.95
0:100:0:0	0.94	1.00	0.85	0.91
0:0:100:0	0.86	0.78	1.00	0.93
0:0:0:100	0.96	0.90	0.92	1.00
70:10:10:10	0.96	0.93	0.84	0.91
10:70:10:10	0.93	0.97	0.82	0.89
10:10:70:10	0.94	0.96	0.82	0.89
10:10:10:70	0.97	0.91	0.87	0.94
25:25:25:25	0.95	0.95	0.83	0.91

Table 3: Directional transfer similarity of domains.

Source Ratio	Target			
En:Fr:Ja:Zh	En	Fr	Ja	Zh
100:0:0:0	1.00	0.99	0.78	0.80
0:100:0:0	0.99	1.00	0.78	0.80
0:0:100:0	0.85	0.85	1.00	0.96
0:0:0:100	0.85	0.85	0.96	1.00
70:10:10:10	0.95	0.95	0.89	0.88
10:70:10:10	0.91	0.91	0.94	0.93
10:10:70:10	0.84	0.84	0.97	0.96
10:10:10:70	0.84	0.84	0.97	0.96
25:25:25:25	0.87	0.87	0.97	0.95

Table 4: Directional transfer similarity of languages.

C Single-source and Mixed-source Transfer Similarity

In Section 3, the main transfer analysis compares single-category tokenizers. Because the token-level experiments also use mixed known-corpus sources, Table 3 and Table 4 report transfer similarity from both single-source and mixed-source tokenizers to single-category targets. Rows specify the source tokenizer's training mixture, and columns specify the target tokenizer whose ID–ratio density is explained.

The mixed-source profiles support the token-level results in Table 1. First, ID–ratio distributions remain broadly transferable after mixing: most mixed-source similarities are still above 0.8, and many domain similarities are close to or above 0.9. Second, mixed sources retain useful similarity to multiple targets rather than only to a single self-matched target. This helps explain why mixed known-corpus sources are generally more stable for token-level estimation. Third, the same structure as in the main heatmap remains visible: English–French and Japanese–Chinese form stronger language clusters, while Code is the hardest domain target because its ID–ratio distribution differs more from natural language corpora.

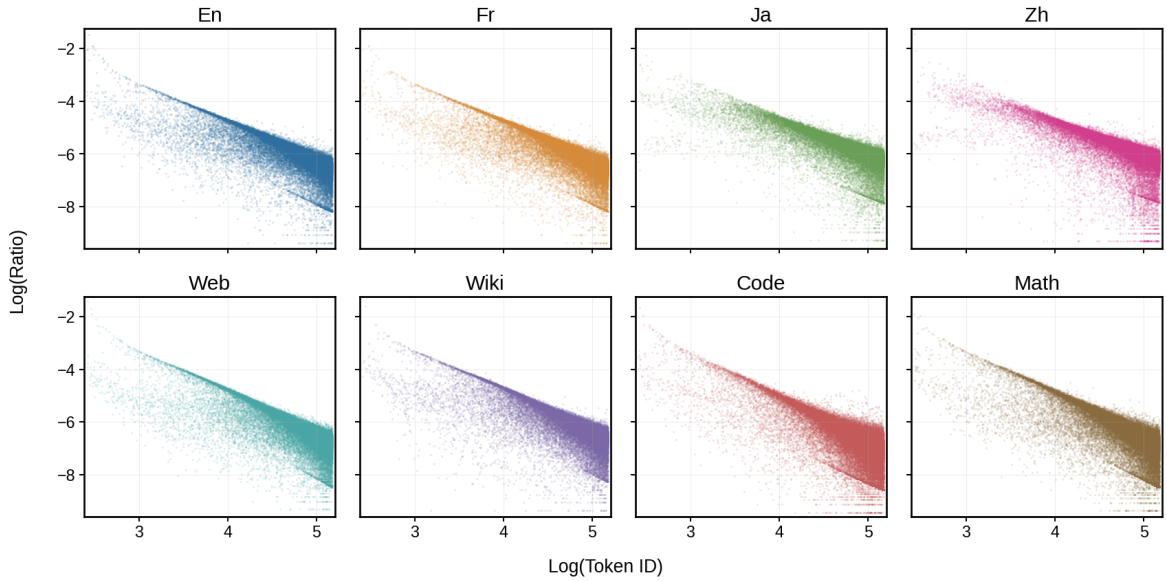

Figure 7: Token ID–ratio scatter plots for all eight controlled tokenizers. The top row shows language-specific mC4 tokenizers, and the bottom row shows domain-specific English tokenizers.

D Additional ID–Ratio Scatter Plots

In Section 3, Figure 2 shows representative ID–ratio scatter plots to motivate transferability. Figure 7 provides the full set of eight controlled tokenizers: the top row shows language-specific mC4 tokenizers, and the bottom row shows domain-specific English tokenizers.

Across both rows, the point clouds follow a similar downward log–log shape, while their thickness, location, and tail behavior vary by language and domain. These patterns support the main-text conclusion that the ID–ratio relationship is broadly shared but not identical across corpora.

E Category-Level Estimation Details

In Section 7, Figure 6 shows selected category-level estimates for the uniform language and domain targets. We present the complete source-composition and K-sweep results in this section, showing that the main conclusions do not depend on a single displayed K or source mixture.

Table 5 follows the same layout as Table 1: single-source columns use one known corpus, 70%-mixed columns use one dominant known corpus with the other three mixed in equally, and the Target-like column uses the same uniform mixture as the target. We report category-level mean relative error, scaled by 100.

QGDE remains below DMI. Across both language and domain targets, QGDE gives substantially lower category-level error than the source-independent DMI baseline. In the language setting, DMI has error 9.09, while the QGDE averages are around 3.1–3.3 across source mixtures. In the domain setting, DMI has error 15.14, while QGDE stays around 5.3–5.5.

Anchor gains are weaker after aggregation. The K-sweep is less monotonic than in token-level estimation. For languages, increasing K improves the early rows but the results quickly cluster near 3.1. For domains, most QGDE rows remain in a narrow band around 5.3–5.5. This supports the observation in Section 7 that category-level aggregation smooths and redistributes token-level improvements, so additional anchors have a weaker visible effect after aggregation.

Language mixtures remain easier. The language block consistently has lower error than the domain block. This matches the main results: language categories provide more separable token-level evidence, whereas domain categories share more vocabulary and therefore make aggregate mixture estimation harder.

F Target-Independent Selection of K

We additionally select K without observing target-corpus ratios. For each known source, we use three folds: two folds fit the QAC trends and select anchors, and the remaining fold measures held-out

Source	Single-source					70%-mixed source					Target-like
	S_1-only	S_2-only	S_3-only	S_4-only	Avg.±Std.	S_1-major	S_2-major	S_3-major	S_4-major	Avg.±Std.	Uniform
Language Sources (S_1 =En, S_2 =Fr, S_3 =Zh, S_4 =Ja)											
DMI	9.09 *(source-independent)*										
QGDE Avg.	**3.25**	**3.21**	**3.22**	**3.14**	**3.21±0.04**	**3.20**	**3.16**	**3.14**	**3.14**	**3.16±0.02**	**3.12**
$K = 3$	3.70	3.58	3.33	3.43	3.51±0.14	3.59	3.47	3.21	3.47	3.44±0.14	3.42
$K = 4$	3.62	3.53	3.26	3.23	3.41±0.17	3.39	3.27	3.16	3.25	3.27±0.08	3.17
$K = 5$	3.43	3.41	3.26	3.17	3.32±0.11	3.31	3.25	3.15	3.14	3.22±0.07	3.11
$K = 6$	3.32	3.31	3.22	3.08	3.23±0.10	3.29	3.21	3.11	3.08	3.17±0.08	3.07
$K = 7$	3.20	3.18	3.19	3.11	3.17±0.03	3.13	3.11	3.14	3.12	3.12±0.01	3.11
$K = 8$	3.18	3.08	3.18	3.11	3.13±0.04	3.15	3.13	3.14	3.09	3.13±0.02	3.10
$K = 9$	3.09	3.01	3.19	3.07	3.09±0.06	3.07	3.08	3.12	3.09	3.09±0.02	3.09
$K = 10$	3.10	3.10	3.18	3.08	3.11±0.04	3.10	3.09	3.09	3.10	3.09±0.00	3.09
$K = 11$	3.11	3.11	3.20	3.07	3.12±0.05	3.12	3.10	3.11	3.08	3.10±0.02	3.04
$K = 12$	3.05	3.04	3.22	3.10	3.10±0.07	3.06	3.05	3.13	3.09	3.08±0.03	3.06
$K = 13$	3.08	3.06	3.23	3.12	3.12±0.07	3.08	3.06	3.15	3.10	3.10±0.03	3.08
$K = 14$	3.17	3.14	3.24	3.13	3.17±0.04	3.12	3.08	3.15	3.11	3.12±0.03	3.09
Domain Sources (S_1 =Web, S_2 =Wiki, S_3 =Math, S_4 =Code)											
DMI	15.14 *(source-independent)*										
QGDE Avg.	**5.44**	**5.37**	**5.40**	**5.34**	**5.39±0.04**	**5.44**	**5.37**	**5.48**	**5.39**	**5.42±0.05**	**5.37**
$K = 3$	5.25	5.44	5.27	5.24	5.30±0.08	5.33	5.40	5.40	5.26	5.35±0.06	5.32
$K = 4$	5.54	5.26	5.39	5.27	5.37±0.11	5.23	5.40	5.42	5.33	5.34±0.08	5.28
$K = 5$	5.50	5.35	5.29	5.27	5.35±0.09	5.60	5.32	5.51	5.32	5.44±0.12	5.29
$K = 6$	5.47	5.47	5.41	5.30	5.42±0.07	5.50	5.36	5.46	5.27	5.40±0.09	5.43
$K = 7$	5.51	5.39	5.35	5.34	5.40±0.07	5.44	5.39	5.51	5.42	5.44±0.05	5.46
$K = 8$	5.53	5.41	5.30	5.34	5.40±0.09	5.46	5.39	5.52	5.43	5.45±0.05	5.39
$K = 9$	5.42	5.30	5.40	5.38	5.37±0.05	5.43	5.39	5.52	5.43	5.44±0.05	5.37
$K = 10$	5.38	5.34	5.42	5.37	5.38±0.03	5.47	5.38	5.54	5.49	5.47±0.06	5.37
$K = 11$	5.35	5.34	5.50	5.38	5.39±0.06	5.47	5.36	5.53	5.43	5.45±0.06	5.36
$K = 12$	5.36	5.35	5.56	5.37	5.41±0.09	5.45	5.35	5.48	5.43	5.43±0.05	5.36
$K = 13$	5.45	5.40	5.48	5.38	5.43±0.04	5.47	5.32	5.45	5.40	5.41±0.06	5.40
$K = 14$	5.48	5.36	5.47	5.44	5.44±0.05	5.49	5.36	5.46	5.43	5.44±0.05	5.41

Table 5: Mean relative error (MRE) (%) of category-level mixture estimation under uniform targets. The source-mixture columns follow the same layout as Table 1. DMI is source-independent and therefore shown as a single merged value. Darker green indicates higher error in the QGDE K-ablation rows.

Selection rule	Overall	Language	Domain	SmolLM
Fixed $K = 14$	4.676%	3.857%	5.087%	6.037%
Held-out selected K	4.413%	3.231%	5.162%	5.858%
Difference	−0.263 pp	−0.626 pp	+0.076 pp	−0.179 pp

Table 6: Error after target-independent K selection.

Proxy corpus	Single-source	70%-mixed	Target-like
mC4	5.80%	4.78%	4.68%
MADLAD-400	4.17%	3.72%	4.02%
Difference	−1.63 pp	−1.06 pp	−0.66 pp

Table 7: Proxy corpus replacement with OSCAR fixed as the target.

error for each $K \in \{3, \ldots, 14\}$. The lowest-error K is then fixed before target evaluation.

As shown in Table 6, held-out selection is slightly better overall and remains comparable to fixed $K = 14$; the domain average is only 0.076 percentage points higher.

G Proxy Corpus Replacement

We keep OSCAR as the target and replace the mC4 proxy used in Table 1 with MADLAD-400.

Table 7 shows that replacing mC4 with MADLAD-400 does not weaken QGDE: error decreases in all three source-setting groups. The SmolLM experiment likewise already uses RedPajama-C4, a Wikipedia–arXiv mixture, and CodeParrot as proxies rather than the exact SmolLM components.

H Additional Validation on Pythia

We use the GPT-NeoX tokenizer released with Pythia (Biderman et al., 2023) and compute reference token ratios from a streaming sample of the deduplicated Pile (Gao et al., 2020). We evaluate QGDE at the target-independent setting $K = 14$ with two mixed-domain proxies: a uniform FineWeb–Wikipedia–OpenWebMath–CodeParrot mixture and a pretrain-like mixture.

As shown in Table 8, QGDE obtains 4.91–4.93% error under the two mixed-domain proxies, compared with 10.42–10.64% for direct transfer and 7.97–8.75% for PoCTrace. This sampled validation supports the token-level result on a second released BPE tokenizer; it does not include a target-like proxy or category-level evaluation.

Source proxy	Transfer	PoCTrace	QGDE
Uniform	10.64	7.97	**4.93**
Pretrain-like	10.42	8.75	**4.91**

Table 8: Token-level validation on Pythia. Values are mean relative log-ratio error (%); lower is better. QGDE uses $K = 14$.

I Tokenizer-Family Scope

Because QGDE relies on BPE merge-order statistics, we test whether the ID–ratio distribution transfers from mC4 BPE tokenizers to OSCAR tokenizers built with BPE, Unigram, or WordPiece (Kudo, 2018; Song et al., 2021).

Table 9 shows that transferability is strongest within BPE, weaker for Unigram, and substantially weaker for WordPiece. QGDE is therefore scoped to BPE in this work.

Transfer setting	En	Fr	Zh	Ja
BPE $\rightarrow$ BPE	0.8778	0.9245	0.9383	0.9579
BPE $\rightarrow$ Unigram	0.7620	0.8574	0.8461	0.8503
BPE $\rightarrow$ WordPiece	0.6421	0.6376	0.5018	0.6167

Table 9: Transferability of token ID–ratio distributions from mC4 to OSCAR.