Shift Is Good: Mismatched Data Mixing Improves Test Performance

Marko Medvedev
University of Chicago

Kaifeng Lyu
Tsinghua University

Zhiyuan Li
TTIC

Nathan Srebro
TTIC

Abstract

We consider training and testing on mixture distributions with different training and test proportions. We show that in many settings, and in some sense generically, distribution shift can be beneficial, and test performance can improve due to mismatched training proportions, even if the components are unrelated and with no transfer between components. In a variety of scenarios, we identify the optimal training proportions and the extent to which such distribution shift can be beneficial. We show how the same analysis applies also to a compositional setting with differing distribution of component "skills" at training and test.

1 INTRODUCTION

Imagine that you are taking a high-stakes exam next week. The exam will be 90% on European history and 10% on Chinese history. Both topics are equally familiar to you and equally difficult, and additional study will help you with each topic similarly. You have unlimited access to study material and practice questions for both. How should you spend your limited studying budget? Should your training match your test distribution, studying 90% European and 10% Chinese? Or would you benefit from a distribution shift? Studying more Chinese history? Less? Only European history? *We encourage the reader to pause and make an intuitive guess.*

The answer depends on the specific learning curve for improvement in test performance within a topic as a function of the number of training examples from that topic. But at least for a generic $1/n$ scaling (as obtained from e.g., both learning VC classes and in parametric regression), the answer, as we will see in Section 3, is that you would benefit from a distribution shift, and should study 75% European History and 25% Chinese history—this would reduce your test error by 20% over the 90%/10% non-shifted training.

We just saw an example of what we term **Positive Distribution Shift**: Even if we have unlimited data from the target test distribution D_{test}, training on a shifted distribution $D_{\text{train}} \neq D_{\text{test}}$ can actually *improve* test performance. This contrasts the typical study of *distribution shift*, i.e., training on one distribution but then applying the predictor, or testing, on another. In that line of work, an implicit baseline would be to train on the test distribution, and any deviation to the case of $D_{\text{train}} \neq D_{\text{test}}$ is viewed as a compromise. This deviation may occur because we do not know or cannot directly access the true D_{test}, because it is too expensive to sample from D_{test}, or because we only have a limited number of samples and want to supplement them with additional data from related distributions. In the standard view, distribution shift is often posed as "how much worse do things get if we train on $D_{\text{train}} \neq D_{\text{test}}$?" A typical answer is of the form: "if D_{train} is close or related enough to D_{test}, then the performance is not much worse."

In this paper, we investigate one of several ways in which distribution shift can be *positive*. We focus on the case when the test distribution is given as a mixture of K components (or tasks), with known mixing proportions $\{p_k\}_{k=1}^K$, and consider training distributions which are mixtures over the same components but with different mixing proportions $\{q_k\}_{k=1}^K$. We study how positive distribution shift can happen *even though the tasks are independent* and there is no transfer, demonstrating that the improvement can arise purely from mixture effects. We systematically demonstrate the benefit of such distribution shift in terms of improved sample complexity when training with mismatched mixing proportions relative to the test distribution. In fact, in Section 7 we argue that positive distribution shift is the norm, rather then the exception, and almost always happens. We can either think of our results as providing guidance when we can actively control mix-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

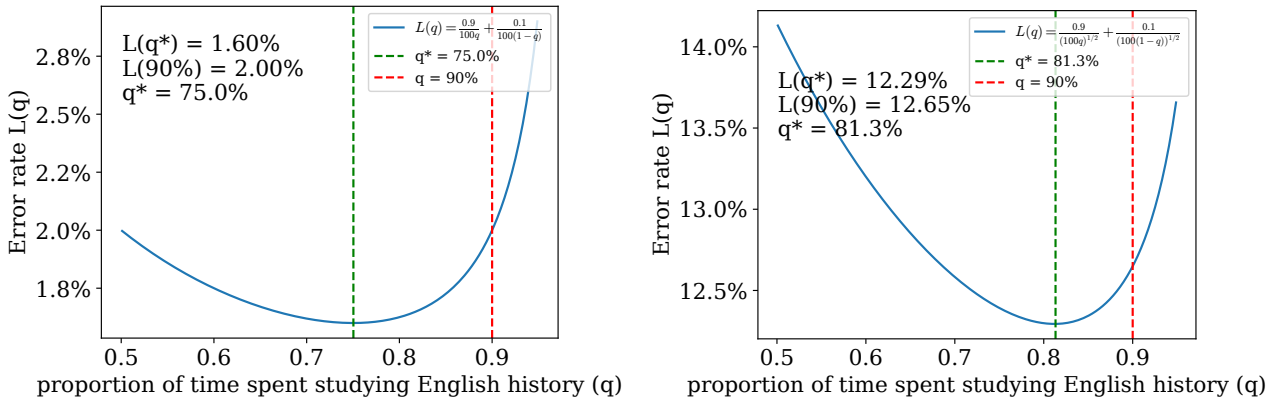

Figure 1: We plot the error rate for a hypothetical scenario modelling the high stakes exam described in Section 1. We model the error rate on each of the test portions as being proportional to $\propto \frac{1}{n_i^\alpha}$, where n_i represents the studying budget spent on that portion of the exam, so $i = 1$ corresponds to European History and $i = 2$ to the Chinese History and set $n_1 + n_2 = N$ to be the total studying budget, with $N = 100$ hours. The exponent α is $\alpha = 1$ on the left plot and $\alpha = 2$ on the right plot. In both cases, we consider $n_1 = qN$ and $n_2 = (1-q)N$, where q is the proportion of time spent studying for the European History portion of the exam. This way, the error rate on the exam can be written as a function of q as $L(q) = 0.9\frac{1}{(100q)^\alpha} + 0.1\frac{1}{(100q)^\alpha}$. We can see on both plots that shifting away from the testing proportion (red line, i.e. $q = 90\%$) can lead to a better error rate with the optimal test proportion (green line, i.e., q^* whose values are displayed accordingly). See also Corollary 3.3.

ing between different known components, or as helping us understand how and why a mismatched training distribution can actually be beneficial.

In Section 5 we go beyond a mixture setting, and consider a compositional problem, where each instance involves composing multiple 'skills', with different skill frequencies. E.g., solving mathematical problems with multiple simple steps, each of which is a 'skill'. Should the training data have the same skill frequencies or different skill frequencies? This problem, which we make concrete as a stylized LLM training problem, was a significant motivator for this research. We show that when training on multiple skills with Chain-of-Thought training, although the setting is different, the effect of changing the skill distribution is related to that of the mixture setting, and thus our mixture setting analysis provides guidance here. We show empirically the benefit of the predicted positive distribution shift in learning this stylized reasoning task.

In Section 6, we depart from components for which learning is independent and consider a setting with *transfer* between the different components. For the most commonly studied transfer learning learning curves, we show that again, even though the error behaviour is different, the effect of mixture proportion mismatch is the same as for the independent learning setting we study, and so the results are applicable also here.

In this paper, we focus on how positive distribution shift can arise purely due to mixture effects, and not because of "transfer" between components or components being more or less informative or useful. Positive distribution shift can certainly arrise also for other reason: E.g., it might be better to train on cleaner or less noisy data, or more generally leveraging transfer patterns between tasks or components Albalak et al. (2023); Liu et al. (2025); Jiang et al. (2025); Shukor et al. (2025). An even stronger benefit might be computational, where changes in the training distribution provide structure that is easier to exploit computationally, as is hinted by e.g. Abbe et al. (2023); Wang et al. (2025). All of these effects can be compounded with the mixture effect we study here. Indeed, several recent empirical papers looked at optimizing training mixture proportions in order to obtain good performance on a mixutre distribution such as Pile, showing empirically that the optimal training proportions differ from the test proportions Xie et al. (2023a); Ye et al. (2025); Albalak et al. (2023); Jiang et al. (2025); Shukor et al. (2025). As also indicated by the empirical scaling laws uncovered, this is due to a large part due to complex non-symetric transfer paterns between the different components, which could give rise to arbitrary Positive Distribution Shift paterns. But in this paper, we focus on understanding and mathematically charactarizing the direct effect of changing the mixture proportions, both as an important effect in their own right, and to better disentangle them from other effects when understanding Positive Distribution Shift forces in more complex problems involving also transfer and computational aspects. The papers mentioned here, as well as others González and Abu-Mostafa (2015); Hoffmann et al. (2022); Sorscher et al. (2022); Xie et al. (2023b); Gu et al. (2025), also emphasize the prevelance of Positive Distribution Shift in practice and how such "data set selection" is an important part of contemporary

machine learning—it would thus benefit us to better understand and charactarize how and why it can happen and obtain a framework and language for discussing it.

2 SETUP

Learning Setup and Loss Let $\ell(h, \boldsymbol{z})$ be the loss function that describes how well a model h performs on an instance $\boldsymbol{z} \in \mathcal{Z}$. For example, in supervised learning, $\boldsymbol{z}$ can be an input-output pair $(\boldsymbol{x}, y)$, and $\ell(h, \boldsymbol{z})$ can be the prediction error of $h(\boldsymbol{x})$ when y is the ground truth. Or, in next-word prediction, $\boldsymbol{z}$ can be a document and $\ell(h, \boldsymbol{z})$ can be the average cross-entropy loss incurred when h is used to predict each of the next tokens in the document. In any case, given a test distribution D_{test} over $\boldsymbol{z}$, we evaluate the model through the *test loss* $\mathcal{L}_{D_{\text{test}}}(h) := \mathbb{E}_{\boldsymbol{z}\sim D_{\text{test}}}[\ell(h, \boldsymbol{z})]$.

Test Distribution. We consider test distributions that can be written as a mixture of K components $\mathcal{D}_1, \ldots, \mathcal{D}_K$. A mixture $\mathcal{D}_{\boldsymbol{r}} = \sum_k r_k \mathcal{D}_k$ is determined by mixing proportions $\boldsymbol{r} = (r_1, \ldots, r_K) \in \Delta_K$, where $\Delta_K := \{\boldsymbol{r} \in \mathbb{R}^K : \boldsymbol{r} \geq 0,\ \sum_{k=1}^K r_k = 1\}$ denotes the probability simplex. In the rest of the paper, we write $\boldsymbol{p}$ for the mixing proportions of the test distribution, i.e., $D_{\text{test}} = \mathcal{D}_{\boldsymbol{p}}$, so the test loss is $\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}}(h) = \mathcal{L}_{\boldsymbol{p}}(h)$, where here and elsewhere we use the subscript $\boldsymbol{p}$ on the loss to denote the mixture $\mathcal{D}_{\boldsymbol{p}}$.

Learning Algorithm. We consider an abstract "learning algorithm" $\mathcal{A}$ that, given training data (or sequence of training examples) $S \in \mathcal{Z}^N$ of size N, produces a model $\mathcal{A}(S)$. The performance of the model is evaluated with test loss $\mathcal{L}_{\mathcal{D}_{\boldsymbol{p}}}(\mathcal{A}(S))$.

Training Distribution. We consider training on N i.i.d. samples $S \sim \mathcal{D}_{\boldsymbol{q}}^N$ from a mixture $\mathcal{D}_{\boldsymbol{q}}$ consisting of the same K components as the test distribution, but with potentially different mixing proportions $\boldsymbol{q} \in \Delta_K$. For training mixing proportions $\boldsymbol{q}$, we denote $L_N(\boldsymbol{p}, \boldsymbol{q}) = \mathbb{E}_{S\sim\mathcal{D}_{\boldsymbol{q}}^N}[\mathcal{L}_{\boldsymbol{p}}(\mathcal{A}(S))]$ the expected test error on $D_{\text{test}} = \mathcal{D}_{\boldsymbol{p}}$ when training with $D_{\text{train}} = \mathcal{D}_{\boldsymbol{q}}$ (we frequently drop the subscript N if its clear from context). The "non-shifted" expected test loss is then denoted $L_N^{\text{same}}(\boldsymbol{p}) = L_N(\boldsymbol{p}, \boldsymbol{p})$. In contrast, we denote $L_N^*(\boldsymbol{p}) = \min_{\boldsymbol{q}\in\Delta_K} L_N(\boldsymbol{p}, \boldsymbol{q})$ the test error with the best mixing ratios, and $\boldsymbol{q}^*$ the minimizing ratios. When $L^* < L^{\text{same}}$ and so $\boldsymbol{q}^* \neq \boldsymbol{p}$, this means we can benefit from mismatched training. **Our main analysis objective is to charactarize $\boldsymbol{q}^*$, L^* and the improvement over L^{same}.**

We measure the mismatch benefit through the improvement in test error for a fixed data size $L_N^{\text{ratio}} = L_N^*/L_N^{\text{same}}$. Or, we measure the sample complexity $N_\epsilon(\boldsymbol{p}, \boldsymbol{q}) = \min\{N : L_N(\boldsymbol{p}, \boldsymbol{q}) \leq \epsilon\}$ and its improvement $N_\epsilon^{\text{ratio}} := N_\epsilon^*(\boldsymbol{p})/N_\epsilon^{\text{same}}(\boldsymbol{p})$. We use the standard $O(\cdot), \Omega(\cdot), \Theta(\cdot), o(\cdot)$ notations for functions of the data size N when characterizing these quantities, and hide dependence on other parameters.

Specifying the Learning Model The expected test loss $L_N(\boldsymbol{p}, \boldsymbol{q})$, and so $\boldsymbol{q}^*$ and the benefit of mismatch, depend on the data distributions and learning behaviour of the algorithm. We capture these by modeling the *subpopulation error function* (or per-component learning curves) $e_k(n_k)$, i.e., the error on each component $\mathcal{D}_k$ when training with n_k examples. That is, for a vector of sample sizes $\boldsymbol{n} = (n_1, \ldots, n_K) \in \mathbb{Z}_{\geq 0}^K$, denote $\boldsymbol{\mathcal{D}^n} = (\mathcal{D}_1)^{n_1} \times \cdots \times (\mathcal{D}_K)^{n_K}$ the distributions over samples with n_i examples from each component $\mathcal{D}_i$. Then $e_k(n_k) = \mathbb{E}_{S\sim\boldsymbol{\mathcal{D}^n}}[\mathcal{L}_{\mathcal{D}_k}(\mathcal{A}(S))]$. The scalar function $e_k(n_k)$ captures the *learning curve* for each component. We focus on the case when there is no interference (positive or negative) or transfer between tasks[1], i.e. when training one task neither helps nor hurts the others, so each e_k is only a function of n_k. In the next two sections, we consider different learning settings, specified by different types of error functions, and characterize $\boldsymbol{q}^*$ and L^* in terms of the error functions. In Section 6 we also consider a setting with transfer between components.

Datasets and Training Sequences In our analysis, we refer to the training budget N and our learning model specifying learning based on n_k examples per component k. We can think of N and $\boldsymbol{n}$ as specifying the number of training examples, in which case the training complexity is a sample complexity. Or, we can think of N as indicating the number of training steps, and n_k as indicating the number of steps in which an example from component k is used. In this case, training complexity is a measure of training time. Either interpretation is valid. But we should emphasize that we only study a dependence on *how many* examples are used from each component, *not* on the *order* (as in curriculum learning).

Learnabilities and Mixing Ratios. We model learning as a function of the *number* of examples from each component, but for our analysis, it will useful to introduce the function $\bar{e}_{N,k}(\boldsymbol{q}) = \mathbb{E}_{S\sim(\mathcal{D}_{\boldsymbol{q}})^n}[\mathcal{L}_k(\mathcal{A}(S))]$, which captures the expected error on component k with mixing proportions $\boldsymbol{q}$. We will refer to $\bar{e}_k(\boldsymbol{q})$ as the subpopulation error function in terms of the mixture $\boldsymbol{q}$. Since the per-component counts $\boldsymbol{n}$ are multinomial, we have $\bar{e}_N(\boldsymbol{q}) = \mathbb{E}_{\boldsymbol{n}\sim\text{Mult}(\boldsymbol{q},N)}[e(\boldsymbol{n})] \in \mathbb{R}^K$ and $L_N(\boldsymbol{p}, \boldsymbol{q}) = \langle \boldsymbol{p}, \bar{e}_N(\boldsymbol{q})\rangle$. Frequently for large sample size N, $e(\boldsymbol{n}), \boldsymbol{n} \sim \text{Mult}(\boldsymbol{q}, N)$, will concentrate around $e(\boldsymbol{q}N)$, and we will sometimes exploit this in the anal-

[1]When we say there is no interference, it is easiest to think of $\mathcal{D}_i$ as having disjoint support, but we do not formally require this as we treat $\mathcal{D}_i$ abstractly and only model the error functions.

ysis, or analyze for $\bar{e}_N(\boldsymbol{q}) \approx e(\boldsymbol{q}N)$.

Knowledge of Test Distribution Mixture Proportions at Test Time. Our main motivation, and the main way to interpret our work, is addressing how could we have positive distribution shift in the mixture setting, i.e., how could $D_{\text{train}} \neq D_{\text{test}}$ be better than $D_{\text{train}} = D_{\text{test}}$, and what qualitative changes in D_{train} make it better. This provides guidance in understanding positive distribution shift and seeking good training distributions. Nevertheless, there are also several realistic examples where it is conceivable the test mixing proportions are known and the training mixing proportions can be controlled or specified. For example, if we are pretraining a large language model for a set of tasks, we might well know their frequency at test time. Even in settings where the exact mixing proportions are unknown, it suffices to estimate the mixing proportions roughly by a quick analysis of test samples. If we suspect a mixture structure in the data, we can try to build crude classifiers for the components. This can be easy for a variety of tasks. For example, in a language-related task if each mixture component is a different language, or in a memorization task, if each mixture component is a topic or area (e.g., sports, science, etc.), we can build crude topic classifiers based on a small amount of data for each topic, or perhaps unsupervised clustering of a sample, then classify a sample. We can estimate the unknown mixing proportions using this classifier. Note that the classifier does not have to be very accurate since we do not care about individual errors, just about the aggregate proportions, and a bit of an error on the proportions is fine, and we demostrate this in Section 3. Similar approach is taken in Ye et al. (2025), where the authors propose mixture dependent scaling laws for finding good training mixture proportion and assume that the validation data comes from an unknown mixture, which they estimate as part of their procedure in finding the good mixture proportions.

3 POWER LAW

Many machine learning tasks can be captured with power law error functions. Some classic examples include linear regression or learning VC classes, both of which have error rate $\propto \frac{1}{N}$, where N is the number of data samples (Shalev-Shwartz and Ben-David, 2014). More recently, there have been many papers studying the loss curves of large language models for different tasks as a function of the compute budget or the number of training tokens through various scaling laws, which model error as having power law dependence on the number of data samples, such as the Chinchilla Scaling Law (Hoffmann et al., 2022), and many others (Kaplan et al., 2020; Cherti et al., 2023; Ye et al., 2025).

To model these situations, we will first consider a setup where all of the K tasks have subpopulation error functions that follow a simple power law in terms of the number of samples.

Model 3.1 (Power Law Error Tasks)**.** There are K tasks. Each task takes data from one of the K subpopulations $\mathcal{D}_i$ that appear in the test distribution with probability p_i and has subpopulation error functions $e_k(n_k)$ that follow a power law, i.e. $e_k(n_k) = \frac{A_k}{n_k^{\alpha_k} + B_k}$ for some $A_k > 0, B_k > 0$, and $0 < \alpha_k \leq 1$.

In Theorem 3.2, we characterize the test error improvement from the positive distribution shift from optimal data mixing ratios in Model 3.1 when the size of the training data N is large.

Theorem 3.2 (Optimal Data Mixing Ratios For Power Law)**.** *In Model 3.1, if for the exponents it holds that $\alpha_1 = \alpha_2 = \cdots = \alpha_S < \alpha_{S+1} \leq \alpha_{S+2} \leq \cdots \leq \alpha_K$ for some S, then there exist $\varepsilon_1, \varepsilon_2 > 0$ such that for any test data mixing ratio $\boldsymbol{p}$ and any $N > N_0(\{A_i, B_i, \alpha_i, p_i\}_{i=1}^K)$ we have that the following holds*

$$q_i^* = \frac{1}{N^{\frac{\alpha_i - \alpha_1}{\alpha_i + 1}}} \left(\frac{\alpha_i p_i A_i}{\left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}} \right)^{\alpha_1 + 1}} \right)^{\frac{1}{\alpha_i+1}} + o\left(\frac{1}{N^{\frac{\alpha_i - \alpha_1}{\alpha_i+1}}} \right) \tag{1}$$

$$L^{\text{same}}(\boldsymbol{p}) = \frac{1}{N^{\alpha_1}} \sum_{i=1}^S p_i^{1-\alpha_1} A_i + o\left(\frac{1}{N^{\alpha_1 + \varepsilon_1}} \right) \tag{2}$$

$$L^*(\boldsymbol{p}) = \frac{1}{N^{\alpha_1}} \left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_i+1}} \right)^{\alpha_1} \left(\sum_{i=1}^S \frac{(p_i A_i)^{\frac{1}{\alpha_i+1}}}{\alpha_i^{\frac{\alpha_i}{\alpha_i+1}}} \right) + o\left(\frac{1}{N^{\alpha_1 + \varepsilon_2}} \right). \tag{3}$$

Theorem 3.2 shows that in the Power Law Model 3.1, positive distribution shift from optimal data mixing ratios improves the prefactor of the test error dependence on the number of data samples N but does not change the decay rate in terms of N. For the proof of Theorem 3.2 and a more precise statement, including the closed form of $N_0(\{A_i, B_i, \alpha_i, p_i\}_{i=1}^K)$, see Appendix A.1.

Furher, we will show that the improvement for positive distribution shift can have significant implications for making training more data efficient. To do so, we show the improvement from this positive distribution shift on the sample complexity in the case where we have one majority population and $K - 1$ minority populations that all have the same power exponent α. This will also include the test-taking example from Section 1.

Corollary 3.3 (Sample Complexity Improvement From Optimal Data Mixing For General Power Law)**.** *Consider Model 3.1 with $S = K$, i.e. $\alpha_1 = \cdots = \alpha_K = \alpha$, $A_1 = \cdots = A_K = A$, and $B_1 = \cdots = B_k = B$ with $\boldsymbol{p} = (p, \frac{1-p}{K-1}, \ldots, \frac{1-p}{K-1})$. We have that for any $\epsilon > 0$*

$$N_\epsilon^{ratio}(\boldsymbol{p}) \leq (1-p) + 2^{\frac{\alpha+1}{\alpha}} \left(\frac{p}{1-p}\right)^{\frac{1}{\alpha+1}} K^{-\frac{\alpha}{\alpha+1}}.$$

Furthermore, the optimal mixing ratios are given by $q_1^* \propto p^{\frac{1}{\alpha+1}}$ and $q_i^* \propto \left(\frac{1-p}{K-1}\right)^{\frac{1}{\alpha+1}}$ for $i \geq 2$.

Corollary 3.3 demonstrates that if we have one majority population and a number of minority populations, the positive distribution shift from optimal data mixing ratio significantly improves sample complexity. For fixed p, if K is large enough, $N^{\text{ratio}}(\boldsymbol{p})$ will be close to $N^{\text{ratio}}(\boldsymbol{p}) \approx 1 - p < 1$, i.e. we get sample complexity improvement of up to p. For example, for $p = 0.7$, $\alpha = 0.28$, and $K = 100$, for any $\epsilon > 0$, $N_\epsilon^{\text{ratio}}(\boldsymbol{p}) \approx 0.75$, i.e. we achieve the same error with $\approx 25\%$ less data samples. We illustrate this in Figure 2. For the proof of Corollary 3.3, see Appendix A.1.

Furthermore, the test taking example considered in the introduction Section 1 follows from Corollary 3.3, by taking $K = 2$, $\alpha = 1$, and $\boldsymbol{p} = (0.9, 0.1)$ (with any A and taking B much smaller than 1). In particular, this shows that the optimal studying budget allocation is $\boldsymbol{q}^* = (0.75, 0.25)$ and the improvement is $N^{\text{ratio}}(\boldsymbol{p}) = 0.8$. This means that if you study for the exam with the right mixing ratio $\boldsymbol{q}^*$, you would need to study 20% less time to achieve the same score as compared to using the test mixing ratio $\boldsymbol{p}$. Further, taking $\alpha = \frac{1}{2}$ we get the second example on Figure 2. This shows that we indeed get $\boldsymbol{q}^* = (0.812\ldots, 0.188\ldots)$ and $N^{\text{ratio}}(\boldsymbol{p}) = 0.944$.

Error Incurred From Using a Crude Classifier. Here we show that in the Power Law Model 3.1, estimating the test mixing proportions $\boldsymbol{p}$ as $\boldsymbol{p}'$ up to precision ε and finding the optimal mixing proportions using the estimated value $\boldsymbol{p}'$ instead of $\boldsymbol{p}$ offsets the final test performance error at most linearly in ε.

Theorem 3.4 (Error Incurred From Using a Crude Classifier in Power Law)**.** *In Power Law Model 3.1 with $\alpha_1 = \alpha_2 = \cdots = \alpha_K$, if we estimate the test mixing proportions $\boldsymbol{p} = (p_1, \ldots, p_k)$ with accuracy ε, i.e. if we compute the training mixing proportions $\boldsymbol{q}$ using $\boldsymbol{p}'$ with $|p_i' - p_i| \leq \varepsilon$ for all i with $\varepsilon \leq \frac{1}{2}\min_i p_i$, then it holds that $\frac{|L^*(\boldsymbol{p}') - L^*(\boldsymbol{p})|}{L^*(\boldsymbol{p})} \leq \frac{\sum_{i=1}^K (A_i)^{\frac{1}{\alpha+1}} p^{-\frac{\alpha}{\alpha+1}}}{\sum_{i=1}^K (A_i p_i)^{\frac{1}{\alpha+1}}} \varepsilon + O(\varepsilon^2)$.*

Theorem 3.4 shows that in the setting where exact test mixing proportions are unkown, it suffices to estimate them with a crude classifier.

4 MEMORIZATION TASKS

Many machine learning tasks involve memorizing a number of unique atoms, such as training LLMs to answer factual questions or explaining the meaning of words, performing tabular RL, and learning transition functions in automata. Here, the atoms correspond to different facts, answers to questions, word meanings, or states. The loss depends on what fraction of atoms the model memorized. The data distribution in this case corresponds to the training data, i.e., in the case of fact retrieval, question answer, or learning word meanings with an LLMs, the data distribution corresponds to the text used for (pre)training.

To model this, we consider a task of memorizing a number of unique atoms from a dataset of fixed size, where the test distribution is a mixture of the tasks we are trying to memorize. More explicitly, let S be the set of possible atoms to memorize and let $s_1, \ldots, s_k \in S$ be the k atoms we are interested in memorizing. Assume that the learning rule memorizes all atoms it has seen so far, and let M be the set of atoms the model has seen. Let the ith component (or task) be memorizing atom s_i. We incur error 0 if $s_i \in M$ and error 1 if $s_i \in M$.

Model 4.1 (Memorization Tasks)**.** Suppose there are K tasks, each of which is a memorization of a unique atom. The test distribution is a mixture of these K tasks, where the k-th task appears with probability p_k. In this case the subpopulation error functions in terms of $\boldsymbol{n}$ are given by $e_k(n_k) = \mathbf{1}_{\{n_k=0\}}$.

The following theorem characterizes the test error improvement from the positive distribution shift from optimal data mixing ratios in the Memorization Task Model 4.1.

Theorem 4.2 (Optimal Data Mixing Test Error Improvement For Memorization Task)**.** *In Model 4.1, for all $\boldsymbol{p} \in \Delta^{K-1}$ with $p_1 \geq p_2 \geq \cdots \geq p_K$, the expected loss when training on n samples is given by*

$$L^{\text{same}}(\boldsymbol{p}) = \sum_{k=1}^K p_k (1-p_k)^N \tag{4}$$

$$L^*(\boldsymbol{p}) = (K_N(\boldsymbol{p}) - 1)\delta_N(\boldsymbol{p}) + \sum_{k=K_N(\boldsymbol{p})+1}^K p_k, \tag{5}$$

where $\delta_N(\boldsymbol{p}) \in [p_{K_N(\boldsymbol{p})+1},\ p_{K_N(\boldsymbol{p})})$ and $K_N(\boldsymbol{p})$ is defined as follows:

$$K_N(\boldsymbol{p}) := \max\left\{ s \leq K : \sum_{k=1}^{s-1} (1 - (p_s/p_k)^{\frac{1}{K-1}}) < 1 \right\}.$$

To understand the magnitute of the test error improvement in Theorem 4.2, we will assume that the test

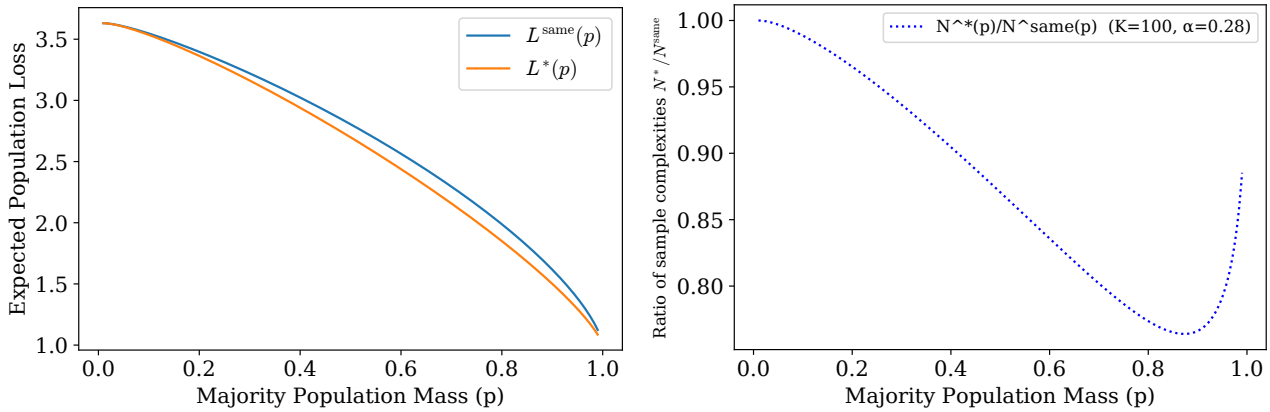

Figure 2: We consider the setup of Corollary 3.3 with $A = 1$, $\alpha = 0.28$, $K = 100$, and some fixed N. On the left plot, we show the "non-shifted" expected population loss $L^{\text{same}}(\boldsymbol{p})$ and the optimally mixed expected population loss $L^*(\boldsymbol{p})$ as a function of majority population mass p. On the right plot, we show the ratio of sample complexities for any fixed $\epsilon > 0$, $N_\epsilon^{\text{ratio}}(\boldsymbol{p})$ as a function of the mass of the majority population, p. We can see significant improvement in the sample complexity from the positive distribution shift from using optimal mixing ratio, even up to $\approx 25\%$.

proportions $\boldsymbol{p}$ follow a power law $p_k = \Theta(k^{-\alpha})$ for some $\alpha > 1$ and that the number of tasks to memorize K is larger than the size of the training set N. In this case, we show that the improvement from positive distribution shift Theorem 4.2 improves even the test error scaling in terms of N. For the proof of Theorem 4.2, see Appendix A.2.

Corollary 4.3 (Test Error Improvement For Memorization Taks with Power Law Test Mixing Ratios)**.** *If $p_k = \Theta(k^{-\alpha})$ for some $\alpha > 1$ and $K = \Omega(N)$, then*

$$L^{\text{same}}(\boldsymbol{p}) = \Theta(N^{-1+\frac{1}{\alpha}}), \qquad L^*(\boldsymbol{p}) = \Theta(N^{-\alpha+1}).$$

For example, when $\alpha = 1.5$, we have $L^{\text{same}}(\boldsymbol{p}) = \Theta(N^{-1/3})$ and $L^*(\boldsymbol{p}) = \Theta(N^{-1/2})$. For the proof of Corollary 4.3, see Appendix A.2.

5 CONNECTION TO SKILL COMPOSITION

Training language models, especially for reasoning tasks including mathematical reasoning, naturally requires the models to learn multiple independent simple skills, and then compose these skills when solving a problem. In this setting, the natural distribution of problems induces a natural distribution of skills at test time. Should we always train the model on the same distribution of skills as the test skill distribution? This is not a mixture-model per-se: each instance is a problem and thus includes many skills, and so the test distribution is *not* a mixture distribution over problems requiring different skills. Nevertheless, in this section, we show that the answer to this question closely follows for the mixture proportions analysis in the previous sections.

A Stylized Model for Compositional Reasoning. To model the above skill composition scenario, we consider the following setting: for a set of skills $[D] = \{1, \ldots, D\}$, the input space is $\mathbf{x} = (s_0, x_1, \ldots, x_K) \in S \times [D]^K$, where S is a set, and the target $y = f_g(\mathbf{x})$ is specified by $g = (g_1, \ldots, g_D) \in G^D$, where $G \subseteq S^S$, as $f_g(\mathbf{x}) = (g_{x_1} s_0, g_{x_2} g_{x_1} s_0, \ldots, g_{x_K} \ldots g_{x_1} s_0)$. The hypothesis class is $\mathcal{H} = \{f_g : S \times [D]^K \to S^K \mid g \in (S^S)^D\}$ and the loss is $\ell(y, \hat{y}) = \mathbf{1}_{y \neq \hat{y}}$. We assume a problem distribution $\mathbf{x} \sim \mathcal{D}_P = \mathcal{D}_S \times \mathcal{D}_X$

A language model h is trained to solve these problems with Chain-of-Thought (CoT) reasoning. After reading the problem $\mathbf{x} = (s_0, x_1, x_2, \ldots, x_K)$, it attempts to apply the skills in order, generating

$$a_0 = s_0,$$
$$a_i \sim P_h(a_i \mid \mathbf{x}, a_0, \ldots, a_{i-1}), \quad i = 1, \ldots, K,$$

where P_h is the sequence distribution induced by the language model h. The test accuracy is defined as $\text{acc}_{\text{test}}(h) = P_{\mathbf{x} \sim \mathcal{D}_P}[a = a^*]$, where $a \sim P_h(a \mid \mathbf{x})$, and a^* is the ground truth output. Expanding this gives

$$\begin{aligned}\text{acc}_{\text{test}}(h) &= \mathbb{E}_{\mathcal{D}_P}[P_h(a^* \mid \mathbf{x})] \\ &= \mathbb{E}_{\mathcal{D}_P}\left[\prod_{i=1}^{K} P_h(a_i^* \mid \mathbf{x}, a_0^*, \ldots, a_{i-1}^*)\right].\end{aligned}$$

To connect with the settings of learning multiple tasks discussed in previous settings, we now introduce two approximations, each of which amounts to an independence assumption:

1. **Step Locality.** The probability of applying the i-th skill correctly depends only on the skill identity, not on the specific input or the surrounding context: $P_h(a_i^* \mid \mathbf{x}, a_0^*, \ldots, a_{i-1}^*) \approx \tilde{P}_h(x_i)$, where $\tilde{P}_h(x_i)$ is a function that only depends on x_i. This treats skill applications as conditionally independent across steps, once the skill type is fixed.

2. **Skill Independence.** The skills $(x_1, \ldots, x_K)$ in $\mathcal{D}_{\mathrm{P}}$ are approximately independent in the problem distribution $\mathcal{D}_{\mathrm{P}}$: $P_{\mathcal{D}_{\mathrm{P}}}(s_0, x_1, \ldots, x_K) \approx P_{\mathcal{D}_S}(s) \cdot \prod_{i=1}^{K} P_{\mathcal{D}_X}(x_i)$.

Under these assumptions, the accuracy simplifies to

$$\mathrm{acc}_{\mathrm{test}}(h) \approx \mathbb{E}_{\mathcal{D}_X}\left[\prod_{i=1}^{K} \tilde{P}_h(x_i)\right] \approx \prod_{i=1}^{K} \mathbb{E}_{x\sim P_{\mathcal{D}_X}}[\tilde{P}_h(x)] = \bar{p}(h)^K,$$

where $\bar{p}(h) := \sum_{x\in[D]} P_{\mathcal{D}_X}(x)\tilde{P}_h(x)$. That is, each skill application succeeds independently with probability $\bar{p}(h)$, which we may view as the model's *per-skill accuracy* averaged under the test distribution. Solving a length-K reasoning chain requires K independent successes, so the overall accuracy scales as $\bar{p}(h)^K$.

Now we take a closer look at the per-skill accuracy $\bar{p}(h)$. This connects directly to our previous multi-task learning framework: each skill $x \in [D]$ corresponds to a distinct task, and the mixing proportion $P_{\mathcal{D}_X}(x)$ represents the natural frequency with which skill x appears in mathematical problems. In this sense, we can set $\boldsymbol{p} \in \Delta_{D-1}$ as a vectorized version of $P_{\mathcal{D}_X}$, and define $\mathcal{L}_{\boldsymbol{p}}(h) = 1 - \bar{p}(h)$ as the per-skill test error. Under our approximations, maximizing the overall test accuracy $\mathrm{acc}_{\mathrm{test}}(h) \approx \bar{p}(h)^K$ is equivalent to minimizing the per-skill test error $\mathcal{L}_{\boldsymbol{p}}(h)$ under the natural frequency of each skill.

Is it always good to train a model on the same per-skill distribution? From the insights we obtained from the previous sections, we see that the best strategy may be to shift the per-skill distribution in a proper way. While the previous sections studied this for abstract learning rules, we will demonstrate this empirically in the next part.

Transformer Experiments. We consider a concrete synthetic task on skill composition. There are D skills, where the i-th skill is a function g_i that maps a number from $\Omega := \{0, \ldots, 9\}$ to Ω. Each skill has a unique English ID. Assume that all these skills are randomly sampled: the IDs are uniformly random from a ID set, and each g_i is uniformly random among all possible functions that map from Ω to Ω. At inference time, a set of K skills $x_1, \ldots, x_K$ are sampled IID following a power law with exponent $\alpha = 1.5$. That is, $\Pr[x = i] \sim i^{-1.5}$ for $i = 1, \ldots, D$. The language model is prompted with the IDs of these skills and a number $s \in \Omega$: "`[s] -> [skill ID 1] -> [skill ID 2] -> ⋯ -> [skill ID k]`". The model is expected to output the result after function composition: $y = g_{x_k}(g_{x_{k-1}}(\cdots g_{x_1}(s)\cdots))$.

Let D_{test} be the distribution of the above prompt and a CoT calculating the correct answer, with $D = 10^5$, K sampled uniformly from 10 to 50. Is the best strategy just training on the same distribution ($D_{\mathrm{train}} = D_{\mathrm{test}}$)? Inspired by our calculation for the memorization task above, properly adjusting the occurrence probability for each skill may lead to better test accruacy. To demonstrate this, we construct another distribution $\mathcal{D}_{\mathrm{uniform}}$ consisting of strings in the form of "`[s] [skill ID] = [expected output]`", where the skill ID and input number are uniformly sampled. In Figure 3, we conduct experiments with a model with GPT-2 architecture and ∼50M parameters. We show that training with $D_{\mathrm{train}} = 30\% \cdot \mathcal{D}_{\mathrm{uniform}} + 70\% \cdot D_{\mathrm{test}}$ significantly outperform training with D_{test}, with around 2.5× speedup in sample efficiency. We defer the experiment details to Appendix D.

6 TRANSFER LEARNING

In this section, we also consider a setting where the tasks are not independent, and there is transfer. Generally, with transfer and in multitask learning, the error on the k-th task is affected by the number of samples on other tasks, that is, the error on task k decreases if we hold n_k constant but increase the number of samples on related tasks. In the most typical transfer learning setups studied in the literature, such as multi-task learning of linear classifiers over linear representation with feature learning (Baxter, 2011; Maurer, 2009; Pontil and Maurer, 2013; Aliakbarpour et al., 2024) and multi-task learning with shared sparsity (Wang et al., 2016, 2017), the transfer effect is captured by the following model, with a slight extension of our framework.

Model 6.1 (Standard Transfer Learning Model)**.** There are K subpopulations, each of which appears in the test distribution with proportion p_k. We extend our framework to allow the subpopulation error functions to depend on all of $\boldsymbol{n}$. Then, for the standard transfer learning model, let $e_k(\boldsymbol{n}) = \frac{A_{0,k}}{(n_1+\cdots+n_k)^{\alpha_k}+B_{0,k}} + \frac{A_{1,k}}{n_k^{\alpha_k}+B_{1,k}}$, for some $A_{0,k}, A_{1,k}, B_{0,k}, B_{1,k} > 0$ and $0 < \alpha_k \le 1$.

For example, in multi-task learning of shared sparsity (Wang et al., 2017), the error bound takes this form with $\alpha_1 = \cdots = \alpha_K = 1$. Interestingly, it turns out that the behavior of the test error in these typical transfer learning settings is the same with respect to the mixture proportions as for the independent component setting, and so our analysis is actually applicable to this non-independent setting as well. The Standard Transfer Learning Model 6.1 is equivalent to the setup of Power Law Tasks Model 3.1 in the sense that we can understand the optimal data mixing ratio $\boldsymbol{q}^*$ and the error improvement of the Standard Transfer Learning model from a specific instance

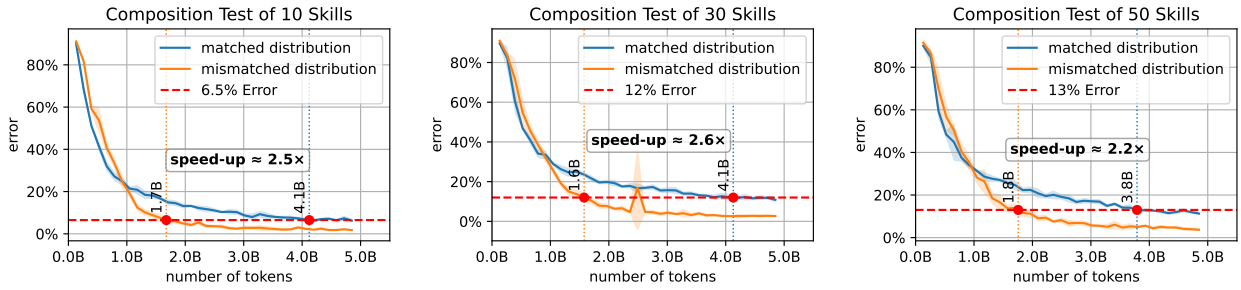

Figure 3: Mismatched distribution improves the test accuracy of a language model in solving a synthetic skill composition task (Section 5). During test, the model is asked to compose several functions, sampled following a power law. Instead of training directly on this task (blue curve), mixing with another task that uniformly samples the functions improves the final accuracy (orange curve). Curves are averaged over 5 random seeds.

of the Power Law Model 3.1. Namely, the transfer term in each of the subpopulation loss functions can be decomposed into a transfer error term and a specific task error term $e_k(\boldsymbol{n}) = e_k^{\text{transfer}}(\boldsymbol{n}) + e_k^{\text{spec}}(\boldsymbol{n})$, where $e_k^{\text{transfer}}(\boldsymbol{n}) = \frac{A_{0,k}}{(n_1+\cdots+n_k)^{\alpha_k}+B_{0,k}}$ is independent of the distribution of samples across different tasks, and $e_k^{\text{spec}}(\boldsymbol{n}) = e_k^{\text{spec}}(n_k) = \frac{A_{1,k}}{n_k^{\alpha_k}+B_{1,k}}$ only depends on n_k. Therefore, the transfer error term $e_k^{\text{transfer}}(\boldsymbol{n})$ in each of the subpopulation error functions will only offset the final expected loss $L(\boldsymbol{p},\boldsymbol{q})$ by $\sum_{i=1}^K p_i \frac{A_{0,k}}{N^{\alpha_k}+B_{0,k}}$, which only depends on the total number of samples N. On the other hand, the specific task error terms $e_k^{\text{spec}}(n_k)$ can be thought of as independent (i.e. without transfer) tasks and will behave the same as in Model 3.1. So, for the Standard Transfer Learning Model 6.1, the optimal data mixing ratio $\boldsymbol{q}^*$ and the expected test losses $L^*(\boldsymbol{p})$ and $L^{\text{same}}(\boldsymbol{p})$ are given by Equation (1), Equation (2) and Equation (3) respectively in Theorem 3.2 with A_k being replaced by $A_{1,k}$.

More complex transfer structures, as is likely usually the case in practice, could lead to even stronger positive distribution shifts, depending on how one task informs the other tasks. In this paper we focus on showing how positive distribution shift can arise *even without* such transfer and understanding purely the effects of mixing proportions. Understanding positive distribution shift more broadly in transfer settings requires specific considerations about the specific form and source of transfer, and we indeed hope to work on transfer learning being described in these terms. We emphasize that we are not aware of this type of analysis for transer learning. This is different from typical descriptions of transfer learning, where data from an alternate or surrogate task is seen as a compromise replacement for additional data from the target task.

7 IT'S ALMOST ALWAYS BETTER TO MISMATCH

So far, we have shown the existence of and quantified the positive distribution shift coming from mistmatched test and train data mixing ratios for the cases of power law tasks in Section 3, memorization tasks in Section 4, and standard transfer learning in Section 6. In this section, we will show that a positive distribution shift coming from the mismatched data mixing ratio almost always exists, i.e., it is almost always better to mismatch the training and test distributions: $\boldsymbol{q}^* \neq \boldsymbol{p}$ and $L^*(\boldsymbol{p},\boldsymbol{q}^*) < L^{\text{same}}(\boldsymbol{p})$.

More precisely, we will show that if there is no positive distribution shift, either the test data mixing ratio is on a measure zero set of the simplex or the subpopulation error functions $e_k(n_k)$ have to be all constant functions, which is meaningless. We show this in Corollary 7.4.

Let $\Delta_+^{K-1} := \{\boldsymbol{p} \in \mathbb{R}^K : \boldsymbol{p} > 0,\ |\boldsymbol{p}| = 1\}$ be the probability simplex and its interior, respectively, where $|\boldsymbol{p}| := \sum_{k=1}^K p_k$. We define $f_k(\boldsymbol{p})$ by extending the domain of each $\bar{e}_k(\boldsymbol{p})$ to the set of non-zero, non-negative vectors $\mathbb{R}^K_{\geq 0} \setminus \{\boldsymbol{0}\}$ by defining $f_k(\boldsymbol{p}) := \bar{e}_k(\frac{\boldsymbol{p}}{|\boldsymbol{p}|})$. We further define $L^{\text{same}}(\boldsymbol{p}) := \sum_{k=1}^K p_k f_k(\boldsymbol{p})$, which extends the definition of L^{same} to the set of non-zero, non-negative vectors $\mathbb{R}^K_{\geq 0} \setminus \{\boldsymbol{0}\}$.

Condition 7.1 (Conservation Condition)**.** *For all* $\boldsymbol{p} \in \mathbb{R}^K_{\geq 0} \setminus \{\boldsymbol{0}\}$, $(f_1(\boldsymbol{p}), \ldots, f_K(\boldsymbol{p})) = \nabla L^{\text{same}}(\boldsymbol{p})$.

Theorem 7.2 (Positive Distribution Shift Almost Always Exists For Data Mixing)**.** *For any set of subpopulations* $\mathcal{D}_1, \ldots, \mathcal{D}_K$ *and any learning algorithm* $\mathcal{A}$*, either Condition 7.1 holds, or there exists a zero-measure set* U *on* Δ_{K-1} *such that for all* $\boldsymbol{p} \in \Delta_{K-1} \setminus U$, $L_N^*(\boldsymbol{p}) < L^{\text{same}}(\boldsymbol{p})$.

Theorem 7.2 shows that either $\boldsymbol{p}$ is on a measure zero set U on Δ_{K-1} or the Conservation Condition 7.1 must hold. Next, we show that if the tasks are independent, then the Conservation Condition 7.1 holds only if all of the subpopulation error functions are constants.

Lemma 7.3 (Independent Tasks). *If $K \geq 3$, and if for all $k \in [K]$, $f_k(\boldsymbol{p}) = g_k(\frac{p_k}{|\boldsymbol{p}|})$ for some function g_k, then Condition 7.1 holds if and only if g_k's are all constant functions.*

Theorem 7.2 and Lemma 7.3 together show that positive distirbution shift always exists, unless all the subpopulation error functions are constant.

Corollary 7.4 (Positive Distribution Shift Always Exists). *For any set of $K \geq 3$ subpopulations $\mathcal{D}_1, \ldots, \mathcal{D}_K$ and any learning algorithm $\mathcal{A}$, if there exists subpopulation $k \in [K]$ such that its error function e_k is not a constant functions over $[N]$ where N is the number of total samples then there exists a measure zero set U on Δ_{K-1} such that for all $\boldsymbol{p} \in \Delta_{K-1} \setminus U$ positive distribution shift from data mixing exists in the sense that there is $\boldsymbol{q}^* \neq p$ for which $L_N(\boldsymbol{p}, \boldsymbol{q}) = L^*(\boldsymbol{p}) < L^{\text{same}}(\boldsymbol{p})$.*

For the proofs of Theorem 7.2, Lemma 7.3, and Corollary 7.4, see Appendix C.

8 RELATED WORKS

Distribution Shift That is Not Harmful. The benefits of mismatching the training and test distribution has already been in studied in some settings. González and Abu-Mostafa (2015) demonstrate positive distribution shift in an entirely different setting, namely linear regression problems with generic mismatched training and test distributions. Unlike in our paper, they do not restrict to changing the train distribution only through data mixing, and generally only show the existence of positive distribution shift in linear rergression problems. They are able to charactarize the optimal shift explicitly only in very special cases. Canatar et al. (2021) show how to numerically optimize the training distribution in high-dimensional kernel regression problems. However, they do not characterize the positive distribution shift, but rather only show how to numerically find it for kernel regression. They do not restrict the test distribution to one coming from a data mixture. Since our focus is on mixing proportions, neither of these investigations fit our framework.

Class-Imbalance. Mismatching training and test distributions for better perfomance has also been studied in class-imbalance literature, which studies how skewed class proportions affect learning, and how resampling or reweighing minority or majority classes can improve performance He and Garcia (2009). Related to our setup with $K = 2$ tasks, Weiss and Provost (2001) studies binary classification and what minority and majority class mix for the training distribution is optimal under a fixed training budget and test distribution, and show that optimal training mix can be different from the test mix. While their setup fits our framework, we study positive distribution shift more broadly and focus on distribution shift that reweighs the mixing proportions to understand its effects on PDS.

Data Mixture Selection. There are a number of empirical papers on finding the optimal data mixture ratios. Xie et al. (2023a) and Liu et al. (2025) train a smaller proxy model to find good mixing proportions and use those for training a large model. Albalak et al. (2023) develops a multi-arm bandit algorithm for online optimization of mixing proportions. Ye et al. (2025) and Shukor et al. (2025) propose new scaling laws that depend on the mixture coefficients for determining the optimal mixing ratio for a target domain. Jiang et al. (2025)adaptively select the mixing proportions based on the scaling laws for each domain separately. The papers consider both optimizing mixing proportions so as to optimize test performance on a different target distribution, and in some papers (Xie et al., 2023a; Ye et al., 2025; Albalak et al., 2023; Jiang et al., 2025; Shukor et al., 2025) also to optimize test performance on a target which is a mixture of the training components with some fixed target proportions—as in our setup. All the papers focus on methods for empirically optimizing training proportions, rather than understanding and charactarizing the phenomena. More importnatly, in the settings considered in these papers there is significant transfer between components, which no doubts dominates the positive distribution shift—we consider a 'pure' setting with orthogonal tasks and no transfer to emphasize how Positive Distribution Shift can occur even in such a setting.

9 SUMMARY

In this paper, we investigate one of the several ways in which distribuion shift can be *positive*, in particular focusing on how positive distribution shift can happen due to mismatched training and test mixture proportions in the setting where the test distribution is a mixture of K tasks or components. We specifically consider independent tasks to understand how mismatched proportions themselves lead to positive distribution shift. We show that in this setting, the optimal training distribution is never equal to the test distribution, except for a measure zero set of test distributions or for those satisfying a conservation property that does not generally hold. Furthermore, we consider different per-components learning curves and the possibility of transfer and in all of these cases we characterize the optimal training mixture and the improvement in sample complexity coming from positive distribution shift.

References

Emmanuel Abbe, Elisabetta Cornacchia, and Aryo Lotfi. Provable advantage of curriculum learning on parity targets with mixed inputs. *Advances in Neural Information Processing Systems*, 36:24291–24321, 2023.

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training, 2023. URL https://arxiv.org/abs/2312.02406.

Maryam Aliakbarpour, Konstantina Bairaktari, Gavin Brown, Adam Smith, Nathan Srebro, and Jonathan Ullman. Metalearning with very few samples per task. In Shipra Agrawal and Aaron Roth, editors, *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 46–93. PMLR, 30 Jun–03 Jul 2024. URL https://proceedings.mlr.press/v247/aliakbarpour24a.html.

Jonathan Baxter. A model of inductive bias learning. *CoRR*, abs/1106.0245, 2011. URL http://arxiv.org/abs/1106.0245.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Out-of-distribution generalization in kernel regression. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12600–12612. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/691dcb1d65f31967a874d18383b9da75-Paper.pdf.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 2818–2829. IEEE, June 2023. doi: 10.1109/cvpr52729.2023.00276. URL http://dx.doi.org/10.1109/CVPR52729.2023.00276.

Carlos R. González and Yaser S. Abu-Mostafa. Mismatched training and test distributions can outperform matched ones. *Neural Computation*, 27(2): 365–387, 2015. doi: 10.1162/NECO_a_00697.

Yuxian Gu, Li Dong, Hongning Wang, Yaru Hao, Qingxiu Dong, Furu Wei, and Minlie Huang. Data selection via optimal control for language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dhAL5fy8wS.

Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. doi: 10.1109/TKDE.2008.239.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

Yiding Jiang, Allan Zhou, Zhili Feng, Sadhika Malladi, and J Zico Kolter. Adaptive data optimization: Dynamic sample selection with scaling laws. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=aqok1UX7Z1.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5BjQOUXq7i.

Andreas Maurer. Transfer bounds for linear feature learning. *Machine Learning*, 75:327–350, 2009. URL https://api.semanticscholar.org/CorpusID:14682470.

Massimiliano Pontil and Andreas Maurer. Excess risk bounds for multitask learning with trace norm regularization. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 55–76, Princeton, NJ, USA, 12–14 Jun 2013. PMLR. URL https://proceedings.mlr.press/v30/Pontil13.html.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Mustafa Shukor, Louis Bethune, Dan Busbridge, David Grangier, Enrico Fini, Alaaeldin El-Nouby, and Pierre Ablin. Scaling laws for optimal data mixtures. *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2507.09404.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=UmvSlP-PyV.

Jialei Wang, Mladen Kolar, and Nathan Srerbo. Distributed multi-task learning. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 751–760, Cadiz, Spain, 09–11 May 2016. PMLR. URL https://proceedings.mlr.press/v51/wang16d.html.

Jialei Wang, Mladen Kolar, Nathan Srebro, and Tong Zhang. Efficient distributed learning with sparsity. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3636–3645. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/wang17f.html.

Zixuan Wang, Eshaan Nichani, Alberto Bietti, Alex Damian, Daniel Hsu, Jason D. Lee, and Denny Wu. Learning compositional functions with transformers from easy-to-hard data. In *COLT*, 2025. URL https://arxiv.org/abs/2505.23683.

Gary M. Weiss and Foster Provost. The effect of class distribution on classifier learning: An empirical study. (ML-TR-44), 2001. doi: 10.7282/t3-vpfw-sf95.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 2023a. URL https://arxiv.org/abs/2305.10429.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=uPSQv0leAu.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jjCB27TMK3.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes, we state the settings clearly throughout the paper.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Not Applicable.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes.
   (b) Complete proofs of all theoretical results. Yes, all resuls are fully contained in their statements and definitions of the settings.
   (c) Clear explanations of any assumptions. Yes, we clearly state conditions of when our results apply.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes, in final section of the appendix.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Not Applicable.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes, in final section of the appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Not Applicable.
   (b) The license information of the assets, if applicable. Not Applicable.
   (c) New assets either in the supplemental material or as a URL, if applicable. Not Applicable.
   (d) Information about consent from data providers/curators. Not Applicable.

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable.

A PROOFS OF ERROR RATE IMPROVEMENTS

A.1 Power Law Tasks

Definition A.1 (Approximate Subpopulation Error Function)**.** For Power Law Model 3.1, let $f_k(\boldsymbol{q})$ be *approximate subpopulation error function* defined as

$$f_k(\boldsymbol{q}) = \frac{A_k}{(q_k N)^{\alpha_k} + B_k}.$$

We define the *approximate expected population loss* as

$$\tilde{L}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{K} p_i f_i(\boldsymbol{q}) = \sum_{i=1}^{K} p_i \frac{A_i}{(q_i N)^{\alpha_i} + B_i}. \tag{6}$$

First, we show that for Power Law Model 3.1 and large number of samples N, it is sufficient to optimize over the approximate expected population loss to find $\boldsymbol{q}*$ up to error of the order $\frac{1}{N}$.

Proposition A.2 (Sufficient to Consider Expectation)**.** *For the approximate error function $f_k(\boldsymbol{q})$ in Definition A.1, we have that when Nq_k is large enough,*

$$|f_k(\boldsymbol{q}) - \bar{e}_k(\boldsymbol{q})| \leq \frac{320 A_k}{B_k} \cdot \frac{1}{(Nq_k)^2} + \frac{\alpha_k A_k}{(Nq_k)^{\alpha_k + \frac{1}{4}}} + \frac{\alpha_k A_k}{(Nq_k)^{\alpha_k + \frac{1}{2}}}.$$

Proof of Proposition A.2. Let $g_k(x) = \frac{A_k}{x^{\alpha_k} + B_k}$. Note that for $n_k \sim \text{Binom}(N, q_k)$ we have that $\mu = \mathbb{E}[n_k] = Nq_k$. So, we have that $f_k(\boldsymbol{q}) = g_k(\mu)$ and $\bar{e}_k(\boldsymbol{n}) = \mathbb{E}[g_k(n_k)]$. Note also that on $(0, \infty)$, $g_k(x)$ is twice differentiable with

$$g_k'(x) = -\frac{A_k\, \alpha_k\, x^{\alpha_k - 1}}{\left(x^{\alpha_k} + B_k\right)^2}$$
$$g_k''(x) = \frac{A_k\, \alpha_k (1 - \alpha_k)\, x^{\alpha_k - 2}}{\left(x^{\alpha_k} + B_k\right)^2} + \frac{A_k\, \alpha_k^2\, x^{2\alpha_k - 2}}{\left(x^{\alpha_k} + B_k\right)^3} = \frac{A_k \alpha_k ((1 - \alpha_k) B_k + (\alpha_k + 1) x^{\alpha_k}) x^{\alpha_k - 2}}{(x^{\alpha_k} + B_k)^3}$$

First, we decompose $\mathbb{E}[g_k(n_k) - g_k(\mu)]$ into two parts:

$$\mathbb{E}[g_k(n_k) - g_k(\mu)] = \underbrace{\mathbb{E}\left[(g_k(n_k) - g_k(\mu)) \mathbb{1}_{\{|n_k - \mu| < \mu^{3/4}\}}\right]}_{=:\delta_1} + \underbrace{\mathbb{E}\left[(g_k(n_k) - g_k(\mu)) \mathbb{1}_{\{|n_k - \mu| \geq \mu^{3/4}\}}\right]}_{=:\delta_2}.$$

For δ_2, by the Multiplicative Chernoff bound we have that if $\mu > 1$

$$P\left(|n_k - \mu| \geq \mu^{\frac{3}{4}}\right) \leq 2 \exp(-\frac{\sqrt{\mu}}{2}).$$

Therefore, we have that

$$\mathbb{E}[(g_k(n_k) - g_k(\mu)) \mathbb{1}_{\{|n_k - \mu| \geq \mu^{3/4}\}}] \leq 2 \exp(-\frac{\sqrt{\mu}}{2}) \frac{A_k}{B_k} \leq \frac{320 A_k}{B_k} \frac{1}{\mu^2}.$$

where we used the fact that $|g_k(n_k) - g_k(\mu)| \leq \max_{x \geq 0} g_k(x) \leq \frac{A_k}{B_k}$.

For δ_1, by Taylor's Theorem, there exists $\xi \in (n_k, \mu)$ (or (μ, n_k)) so that

$$g_k(n_k) = g_k(\mu) + g_k'(\mu)(n_k - \mu) + \frac{1}{2} g_k''(\xi)(n_k - \mu)^2.$$

So we have that

$$\begin{aligned}
|\delta_1| &= \left| \mathbb{E}\left[g_k'(\mu)(n_k-\mu)\mathbb{1}_{\{|n_k-\mu|<\mu^{3/4}\}}\right] + \mathbb{E}\left[\frac{1}{2}g_k''(\xi)(n_k-\mu)^2\mathbb{1}_{\{|n_k-\mu|<\mu^{3/4}\}}\right]\right| \\
&\leq \left|\mathbb{E}\left[|g_k'(\mu)|\cdot|n_k-\mu|\cdot\mathbb{1}_{\{|n_k-\mu|<\mu^{3/4}\}}\right]\right| + \frac{1}{2}\left(\sup_{x\in(\mu-\mu^{3/4},\mu+\mu^{3/4})}|g_k''(x)|\right)\cdot\mu^{\frac{3}{2}} \\
&\leq \frac{\alpha_k A_k}{\mu^{\alpha_k+1}}\mu^{\frac{3}{4}} + \frac{\alpha_k(1-\alpha_k)A_k}{\mu^{\alpha_k+2}}\mu^{\frac{3}{2}} + \frac{\alpha_k^2 A_k}{\mu^{\alpha_k+2}}\mu^{\frac{3}{2}} \\
&\leq \frac{\alpha_k A_k}{\mu^{\alpha_k+\frac{1}{4}}} + \frac{\alpha_k A_k}{\mu^{\alpha_k+\frac{1}{2}}}.
\end{aligned}$$

Where we used the following bound on the supremum of $g_k''(x)$: $\left(\sup_{x\in(\mu-\mu^{3/4},\mu+\mu^{3/4})}|g_k''(x)|\right) \leq \frac{A_k\alpha_k\left((1-\alpha_k)B_k+(\alpha_k+1)(\mu-\mu^{3/4})^{\alpha_k}\right)(\mu-\mu^{3/4})^{\alpha_k-2}}{\left((\mu-\mu^{3/4})^{\alpha_k}+B_k\right)^3} \leq \frac{A_k\alpha_k^2\mu^{\alpha_k}\mu^{\alpha_k-2}}{\mu^{3\alpha_k}}$, as long as $\mu-\mu^{3/4}\geq B_k$ and $(\mu-\mu^{3/4})^{\alpha_k}\geq B_k$, which happens for $Nq_k \geq 2\max\{B_k, B_k^{\alpha_k}\}$.

Putting all these together proves the proposition. □

Proposition A.3 (Optimum of the Approximate Power Law)**.** *Let $\tilde{q}^*$ be the minimum of the approximate population loss defined in Equation (6). For $N > N_0(p_i, A_i, B_i, \alpha_i)$, we have that then*

$$\tilde{q}_i^* = \frac{1}{N^{\frac{\alpha_i-\alpha_1}{\alpha_i+1}}}\left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^S(\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{\frac{1}{\alpha_i+1}} + o\left(\frac{1}{N^{\frac{\alpha_i-\alpha_1}{\alpha_i+1}}}\right)$$

$$\tilde{L}(\tilde{\boldsymbol{q}}^*) = \frac{1}{N^{\alpha_1}}\left(\sum_{i=1}^S(\alpha_i p_i A_i)^{\frac{1}{\alpha_i+1}}\right)^{\alpha_1}\left(\sum_{i=1}^S\frac{(p_i A_i)^{\frac{1}{\alpha_i+1}}}{\alpha_i^{\frac{\alpha_i}{\alpha_i+1}}}\right) + O\left(\frac{1}{N^{\alpha_1+\frac{3\alpha_1^2}{2\alpha_1+2}}}\right). \tag{7}$$

Proof of Proposition A.3. We will take N large enough so that we force $\tilde{q}_i^* \neq 0$, which we do as follows. First take

$$r_i = \frac{1}{N^{\frac{\alpha_i-\alpha_1}{\alpha_i+1}}}\left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^S(\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{\frac{1}{\alpha_i+1}}$$

Take

$$\begin{aligned}
\bar{q}_1 &= \left(\frac{(\alpha_1 p_1 A_1)}{\left(\sum_{i=1}^S(\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{\frac{1}{\alpha_1+1}} - \sum_{i=S+1}^K\frac{1}{N^{\frac{\alpha_i-\alpha_1}{\alpha_i+1}}}\left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^S(\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{\frac{1}{\alpha_i+1}} \\
\bar{q}_i &= r_i \quad \text{for } i>1.
\end{aligned}$$

This way $\sum_{i=1}^K\bar{q}_i = 1$. Take N large enough so that $\bar{q}_1\in(0,1)$, i.e.

$$N > \left(2\left(\frac{(\alpha_1 p_1 A_1)}{\left(\sum_{i=1}^S(\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{\frac{-1}{\alpha_1+1}}\left(\sum_{i=S+1}^K\left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^S(\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{\frac{1}{\alpha_i+1}}\right)\right)^{\frac{\alpha_{S+1}+1}{\alpha_{S+1}-\alpha_1}} \tag{8}$$

suffices because $\frac{1}{N^{\frac{\alpha_{S+1}-\alpha_1}{\alpha_{S+1}+1}}} \geq \frac{1}{N^{\frac{\alpha_i-\alpha_1}{\alpha_i+1}}}$ for all $i\geq S+1$. Note that for these $\bar{q}_i$, we have that for all $i>2$

$$f_i(\bar{\boldsymbol{q}}) \leq \frac{1}{N^{\alpha_i\frac{1+\alpha_1}{1+\alpha_i}}}A_i\left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^S(\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{-\frac{\alpha_i}{\alpha_i+1}}.$$

For $i=1$, we have that $\bar{q}_1 \geq \frac{1}{2}\left(\frac{(\alpha_1 p_1 A_1)}{\left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{\frac{1}{\alpha_1+1}}$, so

$$f_1(\bar{\boldsymbol{q}}) \leq A_1 2^{\alpha_1}\left(\frac{(\alpha_1 p_1 A_1)}{\left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{\frac{-\alpha_1}{\alpha_1+1}}.$$

Therefore, we have that for the approximate expected population loss

$$\begin{aligned}\tilde{L}(\bar{\boldsymbol{q}}) &\leq \frac{1}{N^{\alpha_1}} p_1 A_1 2^{\alpha_1}\left(\frac{(\alpha_1 p_1 A_1)}{\left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{\frac{-\alpha_1}{\alpha_1+1}} + \sum_{i=2}^K p_i \frac{1}{N^{\alpha_i \frac{1+\alpha_1}{1+\alpha_i}}} A_i \left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{-\frac{\alpha_i}{\alpha_i+1}} \\ &\leq \frac{2^{\alpha_1}}{N^{\alpha_1}}\left(\sum_{i=1}^K p_i A_i \left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{-\frac{\alpha_i}{\alpha_i+1}}\right).\end{aligned}$$

Therefore, taking N large enough so that $\tilde{L}(\bar{\boldsymbol{q}}) < \min_i\{1, \frac{A_i}{B_i}\}$ shows that $\tilde{L}$ at $\bar{\boldsymbol{q}}$ is smaller than $\tilde{L}$ for any $\boldsymbol{q}$ with one of $\boldsymbol{q_i}=0$. For this it suffices to take

$$N > 2\left(\frac{1}{\min_i\{1, \frac{A_i}{B_i}\}}\left(\sum_{i=1}^K p_i A_i \left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{-\frac{\alpha_i}{\alpha_i+1}}\right)^{-1}\right)^{\frac{1}{\alpha_1}}. \tag{9}$$

Therefore, we have shown that for N larger than the expressions in Equation (8) and Equation (9), $\tilde{\boldsymbol{q}}^*$ has no zero coordinates.

Now we can find $\tilde{\boldsymbol{q}}^*$ inside $(0,1)^K$ using Lagrange multipliers. Note that each f_i is continuously differentiable on $(0,1)^K$, which is an open set containing the feasible set. Note that the constraint now is $\sum_{i=1}^K q_i - 1 = 0$. We have that there exists $\lambda > 0$ such that

$$\begin{aligned}\frac{p_i A_i}{((Nq_i)^{\alpha_i}+B_i)^2}\alpha_i N^{\alpha_i} q_i^{\alpha_i - 1} &= \lambda \\ \sum_{i=1}^K q_i &= 1.\end{aligned}$$

Note that this equation has a unique solution in $(0,\infty)$ for fixed λ since $\frac{A_i}{((Nq_i)^{\alpha_i}+B_i)^2}\alpha_i N^{\alpha_i}$ is a decreasing function and $\lambda q_i^{1-\alpha_i}$ is an increasing function, and for $q_i = 0$ we have that $\frac{A_i}{B_i}^2 \alpha_i N^{\alpha_i} > 0$. Let λ^* and $\tilde{q}_i = \tilde{q}_i(\lambda^*)$ be the unique solution. Note that $\tilde{L}(\tilde{q}^*) \leq \tilde{L}(\bar{q})$ so in particular we have that

$$p_i \frac{A_i}{(Nq_i)^{\alpha_i}+B_i} \leq C\frac{1}{N^{\alpha_1}},$$

where $C = 2^{\alpha_1}\left(\sum_{i=1}^K p_i A_i \left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{-\frac{\alpha_i}{\alpha_i+1}}\right)$. So we have that

$$\begin{aligned}\frac{p_i A_i}{C} N^{\alpha_1} &\leq (Nq_i)^{\alpha_i} + B_i \\ Nq_i &\geq \left(\frac{p_i A_i}{C} N^{\alpha_1} - B_i\right)^{\frac{1}{\alpha_i}}.\end{aligned}$$

Taking

$$N \geq \frac{2CB_i}{p_i A_i}$$

for all i. Therefore, we have that

$$Nq_i \geq \left(\frac{1}{2}\frac{p_i A_i}{C} N^{\alpha_1}\right)^{\frac{1}{\alpha_i}}$$
$$q_i \geq N^{\frac{\alpha_1-\alpha_i}{\alpha_i}}$$

Therefore as long as

$$N > \left(\max_i \{\frac{2B_i C}{p_i A_i}\}\right)^{\frac{1}{\alpha_1}} \tag{10}$$

we have that

$$\begin{aligned}\frac{A_i}{((Nq_i)^{\alpha_i}+B_i)^2} &\geq \frac{p_i A_i}{(Nq_i)^{\alpha_i}}(1-\frac{B_i}{(Nq_i)^{\alpha_i}})^2 \geq \frac{p_i A_i}{(Nq_i)^{\alpha_i}}(1-\frac{2B_i C}{p_i A_i N^{\alpha_1}})^2 \\ &\geq \frac{p_i A_i}{(Nq_i)^{\alpha_i}}(1-\frac{4B_i C}{p_i A_i N^{\alpha_1}})\end{aligned}$$

for

$$N > \max_i \{B_i^{\frac{1}{\alpha_i}}\}. \tag{11}$$

Therefore, the equation

$$\frac{p_i A_i}{((Nq_i)^{\alpha_i}+B_i)^2}\alpha_i N^{\alpha_i} q_i^{\alpha_i-1} = \lambda$$

implies that

$$\frac{p_i A_i}{(Nq_i)^{2\alpha_i}}\alpha_i N^{\alpha_i} q_i^{\alpha_i-1}(1-\frac{4B_i C}{p_i A_i N^{\alpha_1}}) \leq \lambda \leq \frac{p_i A_i}{(Nq_i)^{2\alpha_i}}\alpha_i N^{\alpha_i} q_i^{\alpha_i-1}.$$

Therefore, for all q we have that

$$\left(\frac{p_i A_i \alpha_i}{N^{\alpha_i}\lambda}\right)^{\frac{1}{\alpha_i+1}}\left(1-\frac{4B_i C}{p_i A_i N^{\alpha_1}}\right)^{\frac{1}{\alpha_i+1}} \leq q_i \leq \left(\frac{p_i A_i \alpha_i}{N^{\alpha_i}\lambda}\right)^{\frac{1}{\alpha_i+1}}.$$

Plugging this back into $\sum_{i=1}^K q_i = 1$ we have that for λ it holds that

$$\sum_{i=1}^K \left(\frac{p_i A_i \alpha_i}{N^{\alpha_i}\lambda}\right)^{\frac{1}{\alpha_i+1}}\left(1-\frac{4B_i C}{p_i A_i N^{\alpha_1}}\right)^{\frac{1}{\alpha_i+1}} \leq 1 \leq \sum_{i=1}^K \left(\frac{p_i A_i \alpha_i}{N^{\alpha_i}\lambda}\right)^{\frac{1}{\alpha_i+1}}$$

Therefore, we have that

$$\lambda^* = \frac{1}{N^{\alpha_1}}\left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_i+1}}\right)^{\alpha_1+1} + O(\frac{1}{N^{2\alpha_1}}).$$

From this we can compute that

$$\tilde{q}_i^* = \frac{1}{N^{\frac{\alpha_i-\alpha_1}{\alpha_i+1}}}\left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^S (\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}}\right)^{\frac{1}{\alpha_i+1}} + O\left(\frac{1}{N^{\frac{\alpha_i-\alpha_1+2\alpha_1}{\alpha_i+1}}}\right). \tag{12}$$

This finishes the proof. The lower bound on N i.e. $N_0(p_i, A_i, B_i, \alpha_i)$ is given by the minimum of Equations (8) to (10) and Equation (11). This shows that

$$\tilde{L}(\tilde{\boldsymbol{q}}^*) = \frac{1}{N^{\alpha_1}} \left(\sum_{i=1}^{S} (\alpha_i p_i A_i)^{\frac{1}{\alpha_i+1}} \right)^{\alpha_1} \left(\sum_{i=1}^{S} \frac{(p_i A_i)^{\frac{1}{\alpha_i+1}}}{\alpha_i^{\frac{\alpha_i}{\alpha_i+1}}} \right) + O\left(\frac{1}{N^{\alpha_1 + \frac{2\alpha_1^2}{\alpha_1+1}}} \right)$$

□

Proposition A.4 (Approximate Optimal is Close to Optimal)**.** *Let $\tilde{\boldsymbol{q}}^*$ be the minimum of the approximate population error $\tilde{L}(\boldsymbol{q})$ in Equation (6) and let $\boldsymbol{q}^*$ be the minimum of the loss in Power Law Model 3.1. Then if $N \geq N_1(p_i, A_i, B_i, \alpha_i)$*

$$L(\boldsymbol{q}^*) = \frac{1}{N^{\alpha_1}} \left(\sum_{i=1}^{S} (\alpha_i p_i A_i)^{\frac{1}{\alpha_i+1}} \right)^{\alpha_1} \left(\sum_{i=1}^{S} \frac{(p_i A_i)^{\frac{1}{\alpha_i+1}}}{\alpha_i^{\frac{\alpha_i}{\alpha_i+1}}} \right) + O\left(\frac{1}{N^{\alpha_1 + \frac{2\alpha_1^2}{\alpha_1+1}}} \right)$$

$$|\tilde{q}_i^* - q_i^*| \leq o\left(\frac{1}{N^{\frac{\alpha_i - \alpha_1}{\alpha_i+1}}}\right)$$

$$q_i^* = \frac{1}{N^{\frac{\alpha_i-\alpha_1}{\alpha_i+1}}} \left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^{S} (\alpha_i p_i A_i)^{\frac{1}{\alpha_1+1}}\right)^{\alpha_1+1}} \right)^{\frac{1}{\alpha_i+1}} + o\left(\frac{1}{N^{\frac{\alpha_i-\alpha_1}{\alpha_i+1}}} \right).$$

Proof of Proposition A.4. Note that by Proposition A.2, we have that for $\tilde{\boldsymbol{q}}^*$ defined in Equation (12)

$$\begin{aligned} L(\tilde{\boldsymbol{q}}^*) &\leq \tilde{L}(\tilde{\boldsymbol{q}}^*) + \sum_{k=1}^{K} p_k \left(320 \min\{\frac{A_k}{B_k}, 1\} \frac{1}{(N\tilde{q}_k)^2} + \frac{\alpha_k A_k}{(N\tilde{q}_k)^{\alpha_k + \frac{1}{4}}} + \frac{\alpha_k A_k}{(N\tilde{q}_k)^{\alpha_k + \frac{1}{2}}} \right) \\ &\leq \tilde{L}(\tilde{q}^*) + \frac{C_L}{N^{\frac{\alpha_1+1}{\alpha_i+1}(\alpha_i + \frac{1}{4})}} \leq \tilde{L}(\tilde{q}^*) + \frac{C_L}{N^{\alpha_1 + \frac{1}{4}}}, \end{aligned}$$

where $C_L = 320 \min\{\frac{A_k}{B_k}, 1\} + 2\alpha_k A_k$. Note additionally that by analogous logic from Proposition A.2 the inequality also holds the other way. By Proposition A.2, we have that

$$L(\boldsymbol{q}^*) \geq \tilde{L}(\boldsymbol{q}^*) - \sum_{k=1}^{K} p_k \left(320 \min\{\frac{A_k}{B_k}, 1\} \frac{1}{(N\tilde{q}_k)^2} + \frac{\alpha_k A_k}{(N\tilde{q}_k)^{\alpha_k + \frac{1}{4}}} + \frac{\alpha_k A_k}{(N\tilde{q}_k)^{\alpha_k + \frac{1}{2}}} \right).$$

Note that since $\tilde{L}(\tilde{\boldsymbol{q}}^*)$ is the minimum, we have that

$$\tilde{L}(\boldsymbol{q}^*) \geq \tilde{L}(\tilde{\boldsymbol{q}}^*).$$

Therefore, we conclude

$$L(\boldsymbol{q}^*) \geq \tilde{L}(\tilde{\boldsymbol{q}}^*) - \sum_{k=1}^{K} p_k \left(320 \min\{\frac{A_k}{B_k}, 1\} \frac{1}{(N\tilde{q}_k)^2} + \frac{\alpha_k A_k}{(N\tilde{q}_k)^{\alpha_k + \frac{1}{4}}} + \frac{\alpha_k A_k}{(N\tilde{q}_k)^{\alpha_k + \frac{1}{2}}} \right).$$

This fnishes the proof of the first claim that

$$L(\boldsymbol{q}^*) = \frac{1}{N^{\alpha_1}} \left(\sum_{i=1}^{S} (\alpha_i p_i A_i)^{\frac{1}{\alpha_i+1}} \right)^{\alpha_1} \left(\sum_{i=1}^{S} \frac{(p_i A_i)^{\frac{1}{\alpha_i+1}}}{\alpha_i^{\frac{\alpha_i}{\alpha_i+1}}} \right) + O\left(\frac{1}{N^{\alpha_1 + \frac{2\alpha_1^2}{\alpha_1+1}}} \right).$$

Note that the above equations imply that

$$|\tilde{L}(\tilde{\boldsymbol{q}}^*) - \tilde{L}(\boldsymbol{q}^*)| \leq 2 \sum_{k=1}^{K} p_k \left(320 \min\{\frac{A_k}{B_k}, 1\} \frac{1}{(N\tilde{q}_k)^2} + \frac{\alpha_k A_k}{(N\tilde{q}_k)^{\alpha_k + \frac{1}{4}}} + \frac{\alpha_k A_k}{(N\tilde{q}_k)^{\alpha_k + \frac{1}{2}}} \right) \leq 2 \frac{C_L}{N^{\alpha_1 + \frac{1}{4}}}.$$

Note now that for all k we have that for all $q, q+h \in (0,1)$ that there is $\xi \in (q+h, q)$ with

$$f_k(q+h) - f_k(q) = f_k'(\xi)h.$$

Therefore, for $k = 1, \ldots, S$ we have that

$$f_i(\tilde{q}_i^* + h) - f_i(\tilde{q}_i^*) = f_i'(\xi_i)h$$

for some $\xi_i \in (\tilde{q}_i^*, \tilde{q}_i^* + h)$. Therefore,

$$|f_i(q_i^*) - f_i(\tilde{q}_i^*)| = |f_i'(\xi_i)||q_i^* - \tilde{q}_i^*|.$$

If for $i = 1, 2, \ldots, S$ we have that $q_i^* > 2\tilde{q}_i^*$, say $i = 1$, there there exists index j such that $q_j^* < \tilde{q}_i^* - \frac{\tilde{q}_1^*}{K}$. Note that all $|f_i'(x)|$ are decreasing, so then $|f_j'(\xi_j)| \geq |f_j'(\tilde{q}_j^*)| = |\lambda| \geq \frac{1}{N^{\alpha_1 + \frac{1}{8}}}$ for N large enough, i.e. it suffices to have

$$N > 16 \left(\sum_{i=1}^{S} (\alpha_i p_i A_i)^{\frac{1}{\alpha_i + 1}} \right)^{-8(\alpha_1+1)}. \tag{13}$$

Then we have that

$$\frac{p_i C_0}{N^{\alpha_1 + \frac{1}{8}}} \leq \frac{p_i}{N^{\alpha_1 + \frac{1}{8}}} |\tilde{q}_i^*| \leq p_i |f_i(q_i^*) - f_i(\tilde{q}_i^*)| \leq |\tilde{L}(\tilde{\boldsymbol{q}}^*) - \tilde{L}(\boldsymbol{q}^*)| \leq \frac{2C_L}{N^{\alpha_1 + \frac{1}{4}}},$$

which is impossible for

$$N > \left(\max_i \{ \frac{2C_L}{p_i} \} \right)^8. \tag{14}$$

Therefore, for all $i = 1, \ldots, S$ we have that $q_i^* \leq 2\tilde{q}_i^*$. Therefore, we have that for all $i = 1, \ldots, S$, $|f_i'(\xi)| \geq \frac{1}{2^{2\alpha_1}} |f_i'(\tilde{q}_i^*)| = \frac{1}{2^{2\alpha_1}} |\lambda| \geq \frac{1}{2^{2\alpha_1}} \frac{1}{N^{\alpha_1}} C_\lambda$, where $C_\lambda = \left(\sum_{i=1}^{S} (\alpha_i p_i A_i)^{\frac{1}{\alpha_i + 1}} \right)^{(\alpha_1 + 1)}$. Therefore, for all $i = 1, \ldots, S$ we have that

$$\frac{1}{2^{2\alpha_1}} \frac{1}{N^{\alpha_1}} C_\lambda |q_i^* - \tilde{q}_i^*| \leq p_i |f_i(q_i^*) - f_i(\tilde{q}_i^*)| |\tilde{L}(\tilde{\boldsymbol{q}}^*) - \tilde{L}(\boldsymbol{q}^*)| \leq 2 \frac{C_L}{N^{\alpha_1 + \frac{1}{4}}}.$$

Therefore, for all $i = 1, \ldots, S$ we have that

$$|q_i^* - \tilde{q}_i^*| < \frac{2^{2\alpha_1 + 1} C_L}{C_\lambda N^{\frac{1}{4}}}.$$

This shows that for $i = 1, \ldots, S$

$$q_i^* = \frac{1}{N^{\frac{\alpha_i - \alpha_1}{\alpha_i + 1}}} \left(\frac{(\alpha_i p_i A_i)}{\left(\sum_{i=1}^{S} (\alpha_i p_i A_i)^{\frac{1}{\alpha_1 + 1}} \right)^{\alpha_1 + 1}} \right)^{\frac{1}{\alpha_i + 1}} + o\left(\frac{1}{N^{\frac{\alpha_i - \alpha_1}{\alpha_i + 1}}} \right).$$

□

Proof of Theorem 3.2. Follows directly from Proposition A.4 and Proposition A.3.

Proposition A.5 (Minimizer is in the interior)**.** *Consider the approximate population error given by Equation (6). Let $\boldsymbol{q}^*$ be the minimum on Δ^{K-1}. Then it holds that $\boldsymbol{q}_i^* \neq 0$ for all i for which $\alpha_i < 1$.*

Proof of Proposition A.5. Assume that $\boldsymbol{q}^*$ is such that $q_i^* = 0$ with $\alpha_i < 1$ for $i \in I$, where I is a set of indices. There exists j with $q_j^* \neq 0$ since $\sum_{i=1}^{K} q_i^* = 1$. Consider the following function

$$g(x) = \sum_{i \in I} \frac{p_i A_i}{(xN)^{\alpha_i} + B_i} + \frac{p_j A_j}{((q_j^* - |I|x)N)^{\alpha_j} + B_j}.$$

Note that

$$g'(x) = -\sum_{i\in I} \frac{p_i A_i}{((xN)^{\alpha_i} + B_i))^2} \alpha_i N^{\alpha_i} x^{\alpha_i - 1} + \frac{p_j A_j}{\left(((q_j^* - |I|x)N)^{\alpha_j} + B_j\right)^2} \alpha_j N^{\alpha_j} (q_j^* - |I|x)^{\alpha_j - 1}.$$

for $x \in (0, \frac{q_j^*}{|I|})$. Note also that on $x \in (0, \frac{q_j^*}{|I|})$, the function $g(x)$ is continous. We have that $\lim_{x\to 0+} g'(x) = -\infty$. There exists $0 < \delta < \frac{q_j^*}{|I|}$ such that $g'(x) < 0$ for all $x \in (0, \delta)$. To see this, not that if this were not the case, there would have to exist a sequence of points $x_1, x_2, \dots$ such that $x_i = 0$.To see why, note that $\lim_{x\to 0} g'(x) = -\infty$ implies that if $g'(x_0) > 0$ then there is $\tilde{x}_0 \in (0, x_0)$ with $g'(\tilde{x}_0) < 0$ and so by IVT we have that there has to exist $x_1 \in (\tilde{x}_0, x_0)$ with $g'(x_1) = 0$. Repeated this procedure gives the sequence $x_1, x_2, \dots$. This is a contradiction since $\lim_{n\to\infty} g'(x_n) = 0$. Therefore, we have that $g(x)$ is decreasing on $(0, \delta)$. Assume that $g(0) \le g(x)$ for all $x \in (0, \delta)$. Therefore, we have that $\frac{g(x)-g(0)}{x} \ge 0$ for all $x \in (0, \delta)$. By MVT, for each $x \in (0, \delta)$ there exists $\xi_x \in (0, x)$ with $g'(\xi_x) = \frac{g(x)-g(0)}{x} \ge 0$. Again, this is a contradiction, since for $x \to 0+$ we have that $\xi_x \to 0+$ so in particular $0 \le \lim_{x\to 0+} g'(\xi_x) = \lim_{x\to 0+} g'(x) = -\infty$. Therefore, there exists $y \in (0, \delta)$ with $g(0) > g(y)$. This contradicts the assumption that $q_i^* = 0$ for all $i \in I$ because if $\tilde{\boldsymbol{q}}_i = \begin{cases} q_i^* & i \ne j, i \notin I \\ y & i \in I \\ q_j^* - |I|y & i = j \end{cases}$, then $\tilde{L}(\tilde{\boldsymbol{q}}, \boldsymbol{p}) < \tilde{L}(\boldsymbol{q}^*, \boldsymbol{p})$. Therefore, $\boldsymbol{q}^*$ has nonzero coordinates for all q_i^* for which $\alpha_i \ne 1$. □

□

Proof of Corollary 3.3. From Theorem 3.2, by directly plugging in we have that since here $S = K$

$$q_i^* = \frac{p_i^{\frac{1}{\alpha+1}}}{\sum_{i=1}^m p_i^{\frac{1}{\alpha+1}}} + o(1)$$

$$q_1^* = \frac{p^{\frac{1}{\alpha+1}}}{p^{\frac{1}{\alpha+1}} + (K-1)\left(\frac{1-p}{K-1}\right)^{\frac{1}{\alpha+1}}} + o(1)$$

$$q_{i\ge 2} = \frac{\left(\frac{1-p}{K-1}\right)^{\frac{1}{\alpha+1}}}{p^{\frac{1}{\alpha+1}} + (K-1)\left(\frac{1-p}{K-1}\right)^{\frac{1}{\alpha+1}}} + o(1)$$

Therefore, this immediately shows the claim about q_i^*. Therefore, we have that

$$N^{\text{ratio}} = \left(\frac{\left(p^{\frac{1}{\alpha+1}} + (K-1)\left(\frac{1-p}{K-1}\right)^{\frac{1}{\alpha+1}}\right)^{\alpha+1}}{p^{1-\alpha} + (K-1)^{\alpha}(1-p)^{1-\alpha}}\right)^{\frac{1}{\alpha}} + o(1).$$

The only thing left to prove is the inequality. Let $\delta = \left(\frac{p}{1-p}\right)^{\frac{1}{\alpha+1}} (K-1)^{-\frac{\alpha}{\alpha+1}}$. Note that we can write

$$p^{\frac{1}{\alpha+1}} + (K-1)\left(\tfrac{1-p}{K-1}\right)^{\frac{1}{\alpha+1}} = (K-1)^{\frac{\alpha}{\alpha+1}}(1-p)^{\frac{1}{\alpha+1}}(1+\delta).$$

Note that $p^{1-\alpha} + (K-1)^{\alpha}(1-p)^{1-\alpha} \ge (K-1)^{\alpha}(1-p)^{1-\alpha}$. Therefore, we can write that

$$N^{\text{ratio}} \le (1-p)(1+\delta)^{\frac{\alpha+1}{\alpha}} + o(1).$$

Note also that $(1+\delta)^t \le 1 + (2^t - 1)\delta$ for $\delta < 1$, since $f(x) = (1+x)^t$ has $f''(x) = t(t-1)(1+x)^{t-2}$ so for $t > 1$ it is convex. Therefore, $f(\delta) \le f(0) + (f(1) - f(0))\delta = 1 + (2^t - 1)\delta$. Using this for $t = \frac{\alpha+1}{\alpha}$, we have that

$$N^{\text{ratio}}(\boldsymbol{p}) \le (1-p) + (2^{\frac{\alpha+1}{\alpha}} - 1)\left(\frac{p}{1-p}\right)^{\frac{1}{\alpha+1}} K^{-\frac{\alpha}{\alpha+1}} + o(1).$$

So it suffices to have the $o(1)$ term be smaller than $\left(\frac{p}{1-p}\right)^{\frac{1}{\alpha+1}} K^{-\frac{\alpha}{\alpha+1}}$. Note that from the proof of Proposition A.4, we can compute the $o(1)$ term. In q_i^*, the term was bounded by $\frac{2^{2\alpha_1+1}C_L}{C_\lambda N^{\frac{1}{4}}}$ where C_L and C_λ can be written explpicitly in terms of A, B, α, K, p. In N^{ratio} the consants are additionally mutliplied by $\frac{1}{\alpha}\left(\frac{\left(p^{\frac{1}{\alpha+1}}+(K-1)\left(\frac{1-p}{K-1}\right)^{\frac{1}{\alpha+1}}\right)^{\alpha+1}}{p^{1-\alpha}+(K-1)^\alpha(1-p)^{1-\alpha}}\right)$, so it suffices to have

$$\frac{1}{\alpha}\left(\frac{\left(p^{\frac{1}{\alpha+1}}+(K-1)\left(\frac{1-p}{K-1}\right)^{\frac{1}{\alpha+1}}\right)^{\alpha+1}}{p^{1-\alpha}+(K-1)^\alpha(1-p)^{1-\alpha}}\right)\frac{2^{2\alpha_1+1}C_L}{C_\lambda N^{\frac{1}{4}}} \le \left(\frac{p}{1-p}\right)^{\frac{1}{\alpha+1}} K^{-\frac{\alpha}{\alpha+1}}$$

$$\frac{1}{\alpha}\left(\frac{\left(p^{\frac{1}{\alpha+1}}+(K-1)\left(\frac{1-p}{K-1}\right)^{\frac{1}{\alpha+1}}\right)^{\alpha+1}}{p^{1-\alpha}+(K-1)^\alpha(1-p)^{1-\alpha}}\right)\frac{2^{2\alpha_1+1}C_L}{C_\lambda N^{\frac{1}{4}}}\left(\frac{p}{1-p}\right)^{-\frac{1}{\alpha+1}} K^{\frac{\alpha}{\alpha+1}} \le N^{\frac{1}{4}}.$$

This happens when $N > N_0(p, \alpha, A, K, B)$. This finishes the proof. □

A.2 Memorization Tasks

Proof of Theorem 4.2. Follows from Lemma A.6 and Lemma A.8. □

Proof Corollary 4.3. First, note that in this case $\left(\frac{p_K}{p_k}\right)^{\frac{1}{N-1}} = \Theta((\frac{k}{K})^{\alpha/(N-1)})$. Therefore, only for $l = \Theta(K)$ do we have $f_N(l) = \Theta(1)$, so indeed then $K_N = \Theta(K)$ in this case. We directly compute that $L^{\text{same}}(\boldsymbol{p}) = \Theta(N^{-1+\frac{1}{\alpha}})$. $L^*(\boldsymbol{p})$ follows directly from Lemma A.8 by using $K_N = \Theta(K)$, and we get $L^*(\boldsymbol{p}) = \Theta(N^{\alpha-1})$

□

Proofs for the Memorization Case For every task k, we only need to memorize the unique hypothesis that appears together with the task.

$$\bar{e}_k(\boldsymbol{q}) = (1-q_k)^N, \qquad L_N(\boldsymbol{p}, \boldsymbol{q}) = \sum_{k=1}^K p_k(1-q_k)^N.$$

Let $\{q_k^*(N)\}_{k=1}^M = \arg\min_{\{q_k\}_{k=1}^M}\{L_N(\boldsymbol{p}, \boldsymbol{q})\}$.

Lemma A.6. *For all $N \ge 1$, there exists $\beta_N > 0$ such that the following holds for $q_k^*(N)$:*

$$q_k^*(N) = \max\left\{0, 1-\beta_N \cdot p_k^{-1/(N-1)}\right\}.$$

Proof. By the method of Lagrange multipliers, there exists $\lambda \in \mathbb{R}$ such that

$$-Np_k(1-q_k^*(N))^{N-1} + \lambda = 0, \qquad \forall k \in [M] \quad \text{s.t.} \quad q_k^*(N) > 0.$$

Then we have

$$q_k^*(N) = 1 - \left(\frac{\lambda}{Np_k}\right)^{1/(N-1)}.$$

Setting $Z_N := \left(\frac{N}{\lambda}\right)^{1/(N-1)}$ and $\beta_N = \frac{1}{Z_N}$ finishes the proof. □

Let $K_N := \max\{k \in [K] : q_k^*(N) \neq 0\}$. K_N and β_N satisfy the following relationship.

Lemma A.7. *For all $N \ge 1$,*

$$\beta_N = \frac{K_N - 1}{\sum_{k=1}^{K_N} p_k^{-1/(N-1)}} \in \left[p_{K_N+1}^{1/(N-1)}, p_{K_N}^{1/(N-1)}\right),$$

$$K_N = \max\{l \mid f_N(l) < 1\} \quad \textit{where} \quad f_N(l) := \sum_{k=1}^{l-1}\left(1 - \left(\frac{p_K}{p_k}\right)^{1/(N-1)}\right).$$

Proof. Since $\sum_{k=1}^{K} q_k^*(N) = 1$ and $q_k^*(N) = 0$ for all $k > K_N$, by Lemma A.6, we have

$$\sum_{k=1}^{K_N} \left(1 - \beta_N \cdot p_k^{-1/(N-1)}\right) = 1.$$

Rearranging the terms, we obtain

$$\beta_N \sum_{k=1}^{K_N} p_k^{-1/(N-1)} = K_N - 1,$$

which implies $\beta_N = \frac{K_N - 1}{\sum_{k=1}^{K_N} p_k^{-1/(N-1)}}$.

By definition of K_N, $1 - \beta_N \cdot p_{K_N}^{-1/(N-1)} > 0$ and $1 - \beta_N \cdot p_{K_N+1}^{-1/(N-1)} \leq 0$. This implies $\beta_N \in [p_{K_N+1}^{1/(N-1)}, p_{K_N}^{1/(N-1)})$. Then we have

$$1 > \sum_{k=1}^{K_N} \left(1 - p_{K_N}^{1/(N-1)} \cdot p_k^{-1/(N-1)}\right) = \sum_{k=1}^{K_N - 1} \left(1 - p_{K_N}^{1/(N-1)} \cdot p_k^{-1/(N-1)}\right).$$
$$1 \leq \sum_{k=1}^{K_N} \left(1 - p_{K_N+1}^{1/(N-1)} \cdot p_k^{-1/(N-1)}\right).$$

Let $f_N(K) := \sum_{k=1}^{K-1} \left(1 - p_K^{1/(N-1)} \cdot p_k^{-1/(N-1)}\right)$. Then $K_N = \max\{K : f_N(K) < 1\}$. □

Lemma A.8. *Test errors for sampling with $\boldsymbol{q} = \boldsymbol{p}$ and $\boldsymbol{q} = \boldsymbol{q}^*$ are*

$$L^{\text{same}}(\boldsymbol{p}) = \sum_{k=1}^{K} p_k (1 - p_k)^N,$$
$$L^*(\boldsymbol{p}) = \sum_{k=K_N+1}^{K} p_k + (K_N - 1)\beta_N^{N-1} \in \left[\sum_{k=K_N+1}^{K} p_k + (K_N - 1)p_{K_N+1}, \sum_{k=K_N+1}^{K} p_k + (K_N - 1)p_{K_N}\right).$$

Proof. The first equation is straightforward. For the second equation,

$$\begin{aligned} L^*(\boldsymbol{p}) = \sum_{k=K_N+1}^{K} p_k + \sum_{k=1}^{N} p_k (1 - q_k^*)^N &= \sum_{k=1}^{K_N} p_k (\beta_N \cdot p_k^{-1/(N-1)})^N \\ &= \sum_{k=K_N+1}^{K} p_k + \beta_N^N \sum_{k=1}^{K_N} p_k^{-1/(N-1)} \\ &= \sum_{k=K_N+1}^{K} p_k + \beta_N^N \cdot \left(\frac{K_N - 1}{\beta_N}\right) \\ &= \sum_{k=K_N+1}^{K} p_k + (K_N - 1)\beta_N^{N-1}. \end{aligned}$$

Further noting that $\beta_N \in \left[p_{K_N+1}^{1/(N-1)}, p_{K_N}^{1/(N-1)}\right)$ completes the proof. □

B INCURRED ERROR USING A CRUDE CLASSIFIER

Here we present a theoretical analysis of how much error do we incurr when using a crude classifier that estimates each test mixing proportion p_i up to some precision ε in the Power Law Model 3.1.

Proof of Theorem 3.4. This follows from direct computation of $|L^*(\mathbf{p}') - L^*(\mathbf{p})|$ using Taylor expansion. In the case that $\alpha_1 = \cdots = \alpha_K$, $L^*(\boldsymbol{p})$ simplifies to

$$L^*(\boldsymbol{p}) = \frac{1}{N^\alpha}\left(\sum_{i=1}^K (A_i p_i)^{\frac{1}{\alpha+1}}\right)^{\alpha+1}.$$

Let $\Delta = \boldsymbol{p}' - \boldsymbol{p}$ and note that $L^*(\boldsymbol{p}') = L^*(\boldsymbol{p}+\Delta)$. Let $f(t) = L^*(\boldsymbol{p}+t\Delta)$. We want to bound $f(1) - f(0)$. Note that L^* is C^2 on $U = \prod_{i=1}^K [p_i/2, 3p_i/2]$, so f is also C^2 on $[0,1]$. Therefore, we can write $f(1) - f(0) = f'(0) + \int_0^1 \int_0^t f''(s)dsdt = f'(0) + \int_0^1 (1-t)f''(t)dt$. Note that $f'(0) = \sum_{i=1}^K \frac{\partial L^*(\boldsymbol{p})}{\partial p_i}\Delta_i$. Further, $f''(t) = \sum_{i,j=1}^K \frac{\partial^2 L^*(\boldsymbol{p}+t\Delta)}{\partial p_i \partial p_j}\Delta_i\Delta_j$, so we have the bound

$$|\int_0^1 (1-t)f''(t)dt| \le \int_0^1 (1-t)\sum_{i,j=1}^K |\frac{\partial^2 L^*(\boldsymbol{p}+t\Delta)}{\partial p_i \partial p_j}||\Delta_i|\Delta_j|.$$

Since L^* is C^2 on U, there is $M > 0$ such that for all $t \in [0,1]$ we have $|\frac{\partial^2 L^*(\boldsymbol{p}+t\Delta)}{\partial p_i \partial p_j}| \le M$. Thereforem we have that $|\int_0^1 (1-t)f''(t)dt| \le MK^2\varepsilon^2$. This implies that

$$L^*(\boldsymbol{p}') - L^*(\boldsymbol{p}) = \sum_{i=1}^K \frac{\partial L^*(\boldsymbol{p})}{\partial p_i}\Delta_i + O(\varepsilon^2).$$

This in turn implies that

$$|L^*(\boldsymbol{p}') - L^*(\boldsymbol{p})| \le \sum_{i=1}^K |\frac{\partial L^*(\boldsymbol{p})}{\partial p_i}|\varepsilon + O(\varepsilon^2).$$

Note that $\frac{\partial L^*(\boldsymbol{p})}{\partial p_i} = \frac{1}{N^\alpha}\left(\sum_{i=1}^K (A_j p_j)^{\frac{1}{\alpha+1}}\right)^\alpha A_i^{\frac{1}{\alpha+1}} p_i^{-\frac{\alpha}{\alpha+1}}$. Plugging this into the above bound finishes the proof. □

C PROOF OF EXISTENCE OF PDS IN THE GENERAL CASE

C.1 Proof of the Main Theorem

We provide a functional-analytic characterization of when positive distribution shift is guaranteed to exist. The key idea is to study the loss $L_N(\boldsymbol{p}, \boldsymbol{r})$ as a function of both the target mixing ratios $\boldsymbol{p}$ and the training mixing ratios $\boldsymbol{r}$, and show that $\boldsymbol{r} = \boldsymbol{p}$ almost never minimizes $L_N(\boldsymbol{p}, \boldsymbol{r})$ except for the degenerate cases described in Theorem 7.2.

The following key property of f_k is useful for our analysis:

Lemma C.1. *For all $k \in [m]$, the function f_k is a 0-homogeneous rational function.*

Proof. It is easy to see that f_k is 0-homogeneous, since by definition, it holds for all $c > 0$ that $f_k(c\boldsymbol{r}) = f_k(\frac{c\boldsymbol{r}}{|c\boldsymbol{r}|}) = f_k(\boldsymbol{r})$.

To show that f_k is a rational function, recall that for $\boldsymbol{r} \in \Delta^{m-1}$, $f_k(\boldsymbol{r})$ is defined as the expected loss of the model trained on a dataset S sampled with mixing ratio $\boldsymbol{r}$ and evaluated on subpopulation $\mathcal{D}_k$.

Sampling from $\boldsymbol{r}$ corresponds to first sampling subpopulation indices $i_1, \ldots, i_n \in [m]$ according to $\boldsymbol{r}$, and then drawing the j-th sample in the dataset S from the subpopulation $\mathcal{D}_{i_j}$. This allows us to rewrite the expectation as:

$$F_k(\boldsymbol{r}) := \sum_{1 \le i_1, \ldots, i_n \le m} \left(r_{i_1} \cdots r_{i_n} \cdot \mathbb{E}_{S \sim \mathcal{D}_{i_1} \times \mathcal{D}_{i_2} \times \cdots \times \mathcal{D}_{i_n}} \mathbb{E}_{\boldsymbol{z} \sim \mathcal{D}_k}\left[\ell(\mathcal{A}(S), z)\right]\right)$$

Each term in this sum $F_k(\boldsymbol{r})$ is the product of a monomial of degree n in $\boldsymbol{r}$ and a constant that does not depend on $\boldsymbol{r}$. Therefore, $F_k(\boldsymbol{r})$ is a degree-n polynomial in $\boldsymbol{r} \in \Delta^{m-1}$. Since $f_k(\boldsymbol{r}) := F_k(\frac{\boldsymbol{r}}{|\boldsymbol{r}|})$ by definition, it follows that f_k is a rational function on $\mathbb{R}^m_{\ge 0} \setminus \{\mathbf{0}\}$. □

Define the total population loss when testing under $\boldsymbol{p}$ but training under $\boldsymbol{r}$ as $L_N(\boldsymbol{p}, \boldsymbol{r}) := \sum_{k=1}^m p_k f_k(\boldsymbol{r})$. We now characterize when $\boldsymbol{r} = \boldsymbol{p}$ is a minimizer of $L_N(\boldsymbol{p}, \boldsymbol{r})$ over $\boldsymbol{r} \in \Delta^{m-1}$.

Lemma C.2. *For any $\boldsymbol{p} \in \Delta_+^{m-1}$, if $L^{\text{same}}(\boldsymbol{p}) = L_N^*(\boldsymbol{p})$, then*

$$\sum_{k=1}^m p_k \frac{\partial f_k(\boldsymbol{p})}{\partial p_i} = 0 \quad \textit{for all } i \in [m]. \tag{15}$$

Proof. We minimize $L_N(\boldsymbol{p}, \boldsymbol{r})$ over $\boldsymbol{r} \in \Delta^{m-1}$ using the method of Lagrange multipliers. Define the Lagrangian:

$$\mathcal{J}(\boldsymbol{r}, \lambda) = \sum_{k=1}^m p_k f_k(\boldsymbol{r}) - \lambda \left(\sum_{k=1}^m r_k - 1 \right).$$

At a minimizer $\boldsymbol{r} = \boldsymbol{p}$, the stationarity condition requires $\frac{\partial}{\partial r_i} \mathcal{J}(\boldsymbol{r}, \lambda) = 0$ for all $i \in [m]$. This yields

$$\frac{\partial}{\partial r_i} \left(\sum_{k=1}^m p_k f_k(\boldsymbol{r}) \right) \Bigg|_{\boldsymbol{r}=\boldsymbol{p}} = \lambda \quad \text{for all } i \in [m].$$

That is,

$$\sum_{k=1}^m p_k \frac{\partial f_k(\boldsymbol{p})}{\partial p_i} = \lambda.$$

Multiplying both sides by p_i and summing over $i \in [m]$ gives:

$$\sum_{i=1}^m p_i \lambda = \sum_{i=1}^m p_i \sum_{k=1}^m p_k \frac{\partial f_k(\boldsymbol{p})}{\partial p_i} = \sum_{k=1}^m \left(p_k \cdot \langle \boldsymbol{p}, \nabla f_k(\boldsymbol{p}) \rangle \right) = \sum_{k=1}^m (p_k \cdot 0) = 0,$$

where the third equality holds because f_k is 0-homogeneous and thus $\langle \boldsymbol{p}, \nabla f_k(\boldsymbol{p}) \rangle = 0$ by Euler's theorem. Thus, $\lambda = 0$, and we have $\sum_{k=1}^m p_k \frac{\partial f_k(\boldsymbol{p})}{\partial p_i} = 0$, as claimed. □

We now connect this condition to a gradient field characterization.

Theorem C.3. *For any learning algorithm $\mathcal{A}$, one of the following two scenarios must hold:*

1. *$L^{\text{same}}(\boldsymbol{p}) = L_N^*(\boldsymbol{p})$ holds only for a zero-measure subset of $\boldsymbol{p} \in \Delta^{m-1}$;*
2. *$\nabla L^{\text{same}}(\boldsymbol{p}) = (f_1(\boldsymbol{p}), \ldots, f_m(\boldsymbol{p}))$.*

Proof. Let Ω_i denote the set of $\boldsymbol{p} \in \Delta_+^{m-1}$ for which the gradient condition (15) holds for index $i \in [m]$. By Lemma C.1, the function f_k is a rational function of $\boldsymbol{p}$. It follows that both $\frac{\partial f_k(\boldsymbol{p})}{\partial p_i}$ and $\sum_{k=1}^m p_k \frac{\partial f_k(\boldsymbol{p})}{\partial p_i}$ are also rational functions of $\boldsymbol{p}$. Therefore, Ω_i is the zero set of a rational function, and must be either a measure-zero subset of Δ_+^{m-1} or the entire domain.

Let $\Omega := \bigcap_{i \in [m]} \Omega_i$ be the intersection of all Ω_i. Then Ω is either a zero-measure subset of Δ_+^{m-1} or the entire domain. If Ω is a zero-measure subset, then by Lemma C.2, we are in the first case of the theorem. If Ω is the entire domain, then the gradient condition (15) holds for all $i \in [m]$, $\boldsymbol{p} \in \mathbb{R}_{\geq 0}^m \setminus \{\boldsymbol{0}\}$.

Recall that $L^{\text{same}}(\boldsymbol{p}) := \sum_{k=1}^m p_k f_k(\boldsymbol{p})$. Then we compute:

$$\frac{\partial L^{\text{same}}(\boldsymbol{p})}{\partial p_i} = f_i(\boldsymbol{p}) + \sum_{k=1}^m p_k \frac{\partial f_k(\boldsymbol{p})}{\partial p_i}.$$

By the gradient condition (15), $\sum_{k=1}^m p_k \frac{\partial f_k(\boldsymbol{p})}{\partial p_i} = 0$. Thus, $\frac{\partial L^{\text{same}}(\boldsymbol{p})}{\partial p_i} = f_i(\boldsymbol{p})$, which implies $\nabla L^{\text{same}}(\boldsymbol{p}) = (f_1(\boldsymbol{p}), \ldots, f_m(\boldsymbol{p}))$, which is the second case of the theorem. □

Finally, Theorem C.3 implies Theorem 7.2.

C.2 Characterization of Conservaation Conditions

Proof of Lemma 7.3. If Condition 7.1 holds, then for all $i, j \in [m]$ $(i \neq j)$,

$$\frac{\partial}{\partial p_j} f_i(\boldsymbol{p}) = \frac{\partial^2}{\partial p_i \partial p_j} L^{\text{same}}(\boldsymbol{p}) = \frac{\partial}{\partial p_i} f_j(\boldsymbol{p}).$$

By the chain rule, we have $\frac{\partial}{\partial p_j} f_i(\boldsymbol{p}) = -\frac{p_i}{|\boldsymbol{p}|^2} g_i'(\frac{p_i}{|\boldsymbol{p}|})$ and $\frac{\partial}{\partial p_i} f_j(\boldsymbol{p}) = -\frac{p_j}{|\boldsymbol{p}|^2} g_j'(\frac{p_j}{|\boldsymbol{p}|})$. Thus, for all $\boldsymbol{p} \in \mathbb{R}^m_{\geq 0} \setminus \{\mathbf{0}\}$,

$$-\frac{p_i}{|\boldsymbol{p}|^2} g_i'(\frac{p_i}{|\boldsymbol{p}|}) = -\frac{p_j}{|\boldsymbol{p}|^2} g_j'(\frac{p_j}{|\boldsymbol{p}|}).$$

For any $x, y > 0$ with $x + y < 1$, we can choose $\boldsymbol{p}$ such that $\frac{p_i}{|\boldsymbol{p}|} = x$ and $\frac{p_j}{|\boldsymbol{p}|} = y$. Then we have $x g_i'(x) = y g_j'(y)$ for all such x, y. This is only possible if there exists a constant C such that $x g_i'(x) = C$ for all $x \in (0, 1)$. Solving this gives $g_i'(x) = \frac{C}{x}$, which implies that $g_i(x) = C \ln x + A$ for some constant A.

Since $g_i(x)$ has no singularity at $x = 0$, we must have $C = 0$. Thus, $g_i(x)$ is a constant function. □

Further, we show that if the Conservation Condition 7.1 is satisfied, then one function f_i determines the rest up to a constant.

Lemma C.4. *If both $(f_1, \ldots, f_K, L^{\text{same}})$ and $(\hat{f}_1, \ldots, \hat{f}_K, \hat{L}^{\text{same}})$ satisfy Condition 7.1, and if $f_i = \hat{f}_i$ for some $i \in [m]$, then for all $k \neq i$, $f_k(\boldsymbol{p}) = \hat{f}_k(\boldsymbol{p}) + C_k$ for some constant C_k.*

The above Lemma C.4 implies that for every k and corresponding error function $e_k(\boldsymbol{n})$, there exists at most one tuple of error functions $\{e_j\}_{j=1, j\neq k}^K$ (up to a individual constant offset for each error function e_j) that positive distribution shift does not happen for $\boldsymbol{p}$ of positive measure. This further implies the following corollary.

Corollary C.5 (Positive Distribution Shift *Almost* Always Exists for General Tasks)**.** *For any set of $K \geq 3$ subpopulations $\mathcal{D}_1, \ldots, \mathcal{D}_K$ and any learning algorithm $\mathcal{A}$, for all $\boldsymbol{p} \in \Delta_+^{K-1}$, the configuration of $[e_k(\boldsymbol{n})]_{k\in[K],\boldsymbol{n}}$ that positive distribution shift does not happen is zero-measure.*

Corollary C.5 shows that either the test mixing ratio $\boldsymbol{p}$ is on a set of measure zero on the simplex or the configuration of subpopulation error functions $e_k(\boldsymbol{n})$ is on a set of measure zero. This implies that positive distribution shift exists *almost* always.

Proof of Lemma C.4. Let $\Delta(\boldsymbol{p}) = L^{\text{same}}(\boldsymbol{p}) - \hat{L}^{\text{same}}(\boldsymbol{p})$ be the difference between the two losses when training on the same distribution. By Condition 7.1, we have

$$\frac{\partial}{\partial p_i} \Delta(\boldsymbol{p}) = \frac{\partial}{\partial p_i} L^{\text{same}}(\boldsymbol{p}) - \frac{\partial}{\partial p_i} \hat{L}^{\text{same}}(\boldsymbol{p}) = f_i(\boldsymbol{p}) - \hat{f}_i(\boldsymbol{p}) = 0. \tag{16}$$

Therefore, $\Delta(\boldsymbol{p})$ is independent of p_i, and there exists a function $C : \mathbb{R}^{m-1} \to \mathbb{R}, \boldsymbol{p}_{-i} \mapsto C(\boldsymbol{p}_{-i})$ such that $\Delta(\boldsymbol{p}) = C(\boldsymbol{p}_{-i})$ for all $\boldsymbol{p} \in \mathbb{R}^m_{\geq 0} \setminus \{\mathbf{0}\}$, where $\boldsymbol{p}_{-i} = (p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_m)$. This is because we can set $C(\boldsymbol{p}_{-i}) = \Delta(\boldsymbol{p})|_{p_i=0}$ and then take the integral of $\frac{\partial}{\partial p_i}\Delta(\boldsymbol{p})$ over p_i to get $\Delta(\boldsymbol{p}) = C(\boldsymbol{p}_{-i})$.

Next, note that both $L^{\text{same}}(\boldsymbol{p})$ and $\hat{L}^{\text{same}}(\boldsymbol{p})$ can be written as rational functions of the form

$$L^{\text{same}}(\boldsymbol{p}) = \frac{S(\boldsymbol{p})}{|\boldsymbol{p}|^n}, \qquad \hat{L}^{\text{same}} = \frac{\hat{S}(\boldsymbol{p})}{|\boldsymbol{p}|^n},$$

where n is the dataset size. This is because $L^{\text{same}}(\boldsymbol{p}) = \sum_{k=1}^m p_k f_k(\boldsymbol{p})$.

Now we show that $\Delta(\boldsymbol{p})$ must have the form $\Delta(\boldsymbol{p}) = \sum_{k\neq i} C_k p_k$ for some constants C_k. Let $D(\boldsymbol{p}) := S(\boldsymbol{p}) - \hat{S}(\boldsymbol{p})$. Since $D(\boldsymbol{p})$ is a polynomial, $C(\boldsymbol{p}_{-i}) = \frac{D(\boldsymbol{p})}{|\boldsymbol{p}|^n}$ must be a rational function. Let $C(\boldsymbol{p}_{-i}) = \frac{A(\boldsymbol{p}_{-i})}{B(\boldsymbol{p}_{-i})}$ for some polynomials $A(\boldsymbol{p}_{-i})$ and $B(\boldsymbol{p}_{-i})$. Then

$$D(\boldsymbol{p}) B(\boldsymbol{p}_{-i}) = A(\boldsymbol{p}_{-i}) |\boldsymbol{p}|^n.$$

If $A = 0$, then $\Delta(\boldsymbol{p}) = 0$. Otherwise, both $A(\boldsymbol{p}_{-i})$ and $B(\boldsymbol{p}_{-i})$ are non-zero polynomials. Since $B(\boldsymbol{p}_{-i})$ cannot be divisible by $|\boldsymbol{p}|^n$, D must be divisible by $|\boldsymbol{p}|^n$. Note that D is a $(n+1)$-homogeneous polynomial and $|\boldsymbol{p}|^n$ is

n-homogeneous, so $C(\boldsymbol{p}_{-i}) = \frac{D}{|\boldsymbol{p}|^n}$ must be a 1-homogeneous polynomial. The only 1-homogeneous polynomials in variables $\boldsymbol{p}_{-i}$ are linear functions of the form $C(\boldsymbol{p}_{-i}) = \sum_{k\neq i} C_k p_k$ for some constants C_k. Thus, no matter $A = 0$ or not, we have $\Delta(\boldsymbol{p}) = \sum_{k\neq i} C_k p_k$.

Finally, by Condition 7.1, we can compute for all $k \neq i$ that

$$f_k(\boldsymbol{p}) - \hat{f}_k(\boldsymbol{p}) = \frac{\partial}{\partial p_k}\Delta(\boldsymbol{p}) = C_k,$$

which implies that $f_k(\boldsymbol{p}) = \hat{f}_k(\boldsymbol{p}) + C_k$, as desired. □

Proof of Corollary C.5. This follows from Lemma C.4. Note that Lemma C.4 implies that for every k and corresponding error function $e_k(\boldsymbol{n})$, there exists at most one tuple of error functions $\{e_j\}_{j=1,j\neq k}^K$ (up to a individual constant offset for each error function e_j) that positive distribution shift does not happen for $\boldsymbol{p}$ of positive measure. This implies the corollary. □

D EXPERIMENT DETAILS

Model Architecture and Tokenizer. We use a model architecture similar to GPT-2, except that we use RoPE instead of absolute position embedding. Our model has 6 layers, 8 attention heads, and 512 embedding dimensions. We use the same tokenizer as GPT-2, which is a byte-pair encoding (BPE) tokenizer.

Generation of Skills. We randomly generate $M = 10^5$ skills. For each skill, we randomly sample 3 English tokens and concatenate them to form the skill ID. The first token is sampled from a set of 1000 tokens that start with a blank space and then a capital letter. The second and third tokens are sampled from a set of 1000 tokens that start with a captial letter without a blank space. The starting blank space is to ensure that the skill ID is tokenized into exactly 3 tokens when placed in a prompt with space-separated skill IDs. For example, "`CourtClientCheck`" can be a skill ID (with blank space removed). Then, for each skill i, we uniformly randomly sample a function g_i that maps a number from $\{0,\ldots,9\}$ to $\{0,\ldots,9\}$.

Distribution: Skill Composition. For each data point, a number k is sampled uniformly from $\{10,\ldots,50\}$, then a set of k skills $g_{i_1},\ldots,g_{i_k}$ are sampled IID following a power law $p(i) \propto (i+50)^{-\alpha}$ with exponent $\alpha = 1.5$. The text consists of two parts. The input part is as follows:

```
<|begin_of_text|> Input:
[x] -> [skill ID 1] -> [skill ID 2] -> ... -> [skill ID k]
```

The output part is as follows:

```
Output:
[x] -> [skill ID 1] = [x1]
[x1] -> [skill ID 2] = [x2]
[x2] -> [skill ID 3] = [x3]
...
[xk-1] -> [skill ID k] = [xk]
[xk]
```

The input and output parts are concatenated together with a blank line in between.

Distribution: Uniform Skills. For each data point, we randomly sample a skill ID uniformly from the skill ID set. Then the text is as follows:

```
<|begin_of_text|> [x] [skill ID] = [expected output]
```

Evaluation. We evaluate the test accuracy of the model on skill composition task with CoT reasoning. We sample 400 data points from the skill composition task, but fix k to be $10, 30, 50$. For each data point, only the input part is given to the model, and the model's autoregressive output is considered as correct if the last line of the output is the same as the expected output.

Training with Matched Distribution. In the training, we use batch size 128 and maximum sequence length 2048. We perform sequence packing for each sequence in the batch: we sample data points from the skill composition task until the maximum sequence length is reached. We train the model on 4 A4000 GPUs for at most 40K steps.

Training with Mismatched Distribution. Similar as above, but for every sequence in the batch, we first choose the task to be skill composition or uniform skills with probability 70% and 30% respectively. Then we sample data points from the chosen task until the maximum sequence length is reached. These sequences are then packed together to form the batch.

Results. We show the results in Figure 3. We see that training with mismatched distribution significantly outperforms training with matched distribution.