UNDERSTANDING INFERENCE SCALING LAWS FOR MIXTURES OF LLMS

Alex Havrilla
School of Mathematics
Georgia Institute of Technology
ahavrilla3@gatech.edu

Srishti Gureja
Cohere
srishtias1016@gmail.coma

ABSTRACT

Scaling inference time compute has enabled a significant improvement in model mathematical problem solving ability. However, most inference scaling strategies sample only from a single model. We extend and analyze inference scaling in the mixed model setting, where samples from weak but inexpensive and strong but expensive models can be pooled at test time. We find mixing samples over a distribution of problems can outperform the best pure, single model strategy by over 5% when given the same compute budget. Further, model mixing extends the compute regimes for which inference scaling reliably improves performance. However, as part of our analysis, we prove that for a **fixed problem** Q a pure strategy sampling only a single model is most efficient. Further, the best model can be identified as having the largest *compute normalized probability* of success for Q. This implies the observed empirical improvements from model mixing stem from an average improvement over the problem distribution as opposed to improvement over the best pure strategy for any single problem. To better understand this result we empirically analyze the distribution of compute normalized probabilities over problems for variously sized models. Our analysis reveals each model is best suited for efficiently solving a non-trivial subset of problems, further motivating the effectiveness of mixing solutions. Somewhat surprisingly, this remains true even for the hardest set of problems, where, for example, the smallest model is most efficient in solving 25% of the problem set.

1 INTRODUCTION

Recently, scaling the inference time compute of reasoning models has emerged as a powerful tool for boosting problem solving ability (Brown et al., 2024; Snell et al., 2024; Wu et al., 2024). RL fine-tuned models like o1 (OpenAI, 2024) and r1 (DeepSeek-AI et al., 2025) exhibit an ability to explore multiple branches of thought, conducting an in-context search for the correct solution. Remarkably, these methods exhibit a log-linear scaling law between inference-time compute and model performance over several orders of magnitude. However, all of these approaches sample solutions only from a single model, thereby potentially limiting solution diversity (Havrilla et al., 2024).

In this work we study the inference time scaling benefits of *model mixing*: allocating a fixed compute budget C amount multiple models $M_1, ..., M_k$ instead of a single model M. We are primarily interested in two research questions (**RQ**s):

- **RQ1:** Can model mixing improve inference time scaling either by 1) boosting performance or 2) **extending** the scaling law to larger compute regimes?
- **RQ2:** If the answer to **RQ1** is yes, **what factors** account for this improvement?

In answer to **RQ1** we find that model mixing both improves overall inference scaling performance and extends the compute regimes for which scaling continues. In an effort to address **RQ2** and understand what factors might account for this improvement, we prove that for a **fixed problem** Q a pure strategy sampling only a single model is most efficient. This implies the observed empirical improvements from model mixing stem from an **average improvement over the problem**

distribution as opposed to improvement over the best pure strategy for any single problem. Our analysis leads us to define the *compute normalized probability* (**cnp**) of success at solving a problem Q as a quantity useful for identifying which problems are most efficiently solved by which models from a set $M_1, ..., M_k$ with costs $c_1, ..., c_k$. Empirically computing the distribution of cnps for $M_1, ..., M_k$ across a large set of reasoning problems reveals each M_i is most efficient for solving a large percentage of problems across all difficulty levels.

In summary we make the following contributions:

- An extension of inference-time scaling to the mixed model setting, revealing benefits to both overall performance and scaling longevity.
- An analysis of factors leading to improvement, revealing large, disjoint subsets of problems across all difficulty levels are most efficiently solved by different models.

Background The inference time scaling benefits of LLMs for reasoning tasks were initially revealed in Brown et al. (2024); Snell et al. (2024); Wu et al. (2024), all of which compared the test performance of LLMs versus compute used when equipped with various search strategies (i.i.d parallel sampling, MCTS (Browne et al., 2012), etc). More recently, o1 (OpenAI, 2024) and r1 (DeepSeek-AI et al., 2025) which train LLMs to conduct an in-context search process, thereby scaling the number of sampled tokens in a given solution attempt. Most related to this work is the recent Zhang et al. (2024) which proposes a training based method for finding the ideal allocation of compute among multiple models to boost inference time performance. Also related are works which combine the usage of small and large models at test time by dynamically selecting the best model on a question by question basis (Shufaro et al., 2024).

2 METHODS

For the remainder of the work we fix our model set M_1, M_2, M_3 as the Qwen-1.5-4B, 14B, and 32B models respectively. We evaluate on reasoning problems drawn from the MATH (Hendrycks et al., 2021) test set. Let $\Delta(3)$ denote the two-dimensional probability simplex such that $\lambda \in \Delta(3)$ satisfies $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Then, given a compute budget $C > 0$, the λ-mixing M_λ of M_1, M_2, M_3 allocates $\lambda_i \cdot C$ compute to M_i. For a distribution of problem solution pairs $(Q, A) \sim P$ we then seek to maximize

$$\lambda^* = \arg\max_{\lambda \in \Delta(3)} \mathop{\mathbb{E}}_{(Q,A)\sim P} \mathbb{P}_{M_\lambda}(A|Q)$$

where λ^* will be our optimal mixture allocation. We know that, for a fixed problem solution pair (Q, A) the optimal allocation is in fact pure (no mixing):

Proposition 1. *Let $k \in \mathbb{N}$ and $M_1, ..., M_k$ be language models with inference costs $c_1, ..., c_k$. Fix a problem, answer pair (Q, A). For $1 \leq i \leq k$, let $p_i = \mathbb{P}_{M_i}(A|Q)$ and $r_i = \frac{\log(1-p_i)}{c_i}$ be the **log-compute normalized failure probability** of model M_i for Q. Fix a compute budget $C > 0$ and $\boldsymbol{M}_\lambda$ the λ ensemble of $M_1, ..., M_k$ for some λ in $\Delta(k)$ where $\Delta(k)$ is the $k-1$ dimensional probability simplex. Then*

$$\arg\max_{\lambda \in \Delta(k)} \mathbb{P}(A|Q, \boldsymbol{M}_\lambda) = \arg\max_{e_i} \mathbb{P}(A|Q, \boldsymbol{M}_\lambda) = e_{i^*}$$

and

$$i^* = \arg\min_{1 \leq i \leq k} r_i$$

where e_i is the ith standard basis vector. I.e. the probability of solving the fixed question Q with compute budget C is maximized by only sampling the model M_i with the smallest compute inefficiency r_i.

A proof is provided in Appendix Section B. The proposition reveals that if model mixing does lead to better inference scaling laws, this cannot be due to an improvement over the best pure strategy for a single question Q. Additionally, the proof of 1 suggests the following useful quantity identifying which model among M_1, M_2, M_3 will most efficiently solve a fixed Q. We define the *compute normalized probability* of success for M_i on Q as $1-(1-p_{i,Q})^{\frac{c_{\max}}{c_i}}$ where $c_{\max} = \max_i c_i$. The quantity

measures the probability model M_i successfully solves Q after $\frac{c_{\max}}{c_i}$ solution attempts (where without loss of generality we suppose $c_i \mid c_{\max}$). We'll now experimentally evaluate a range of M_λ model mixtures and investigate which types of problems are best solved by which models.

3 Results when Scaling Mixed Inference Compute

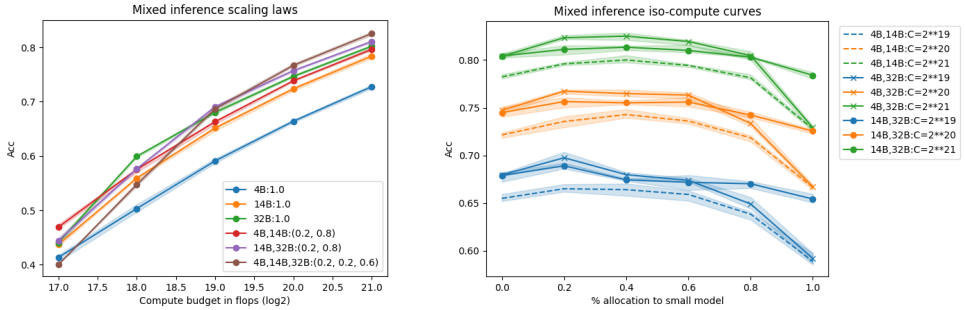

Figure 1: **Left:** Mixed inference compute scaling laws for different λ. **Right:** Performance curves for two-way mixtures for fixed compute budgets when varying the % allocation to the smaller model.

Mixing models leads to more performant scaling laws We now sample solutions from M_1, M_2, M_3 using up to 2^{21} flops per problem. For each compute budget $C \leq 2^{21}$ we run a run a grid search over $\lambda \in \Delta(3)$ with a step-size of 0.1. Figure 1 plots the inference time scaling behavior of selected mixtures. The best performing one-way mixture solves 77% with the full budget. In comparison, a mixture allocating 20% compute to the 4B model, 20% to 14B, and 60% to 32B solves 82% of problems, demonstrating the improvement from model mixing. Additionally, the slope of the mixed model curve is steeper than the one-way curves, indicatin better scalability.

In Figure 1 we additionally plot curves for the two-way mixtures at various compute budgets, varying the % allocation to the smaller model. In general it appears the largest model (32B) performs best even in combination with either of the other model. Somewhat surprisingly, the best performance consistently comes from the combination of the 4B and 32B models, instead of 14B and 32B. This is despite the 4B model performing worst overall with the equal compute budgets. This suggests that while the 32B model is the most compute efficient over the entire distribution of problems, the types of problems it fails to solve are better complemented by 4B than 14B.

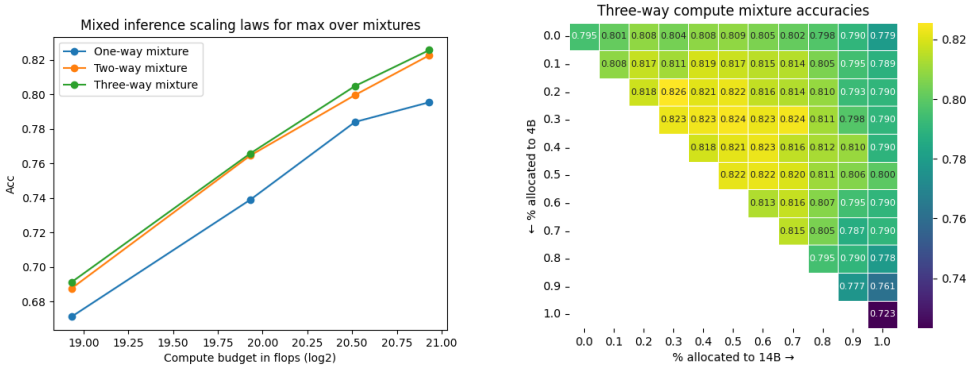

Figure 2: **Left:** Max performance over one, two, and three way mixture strategies vs. compute. **Right:** Three-way mixture performance with 2^{21} flop compute budget.

Three-way mixtures are not overly sensitive to mixture allocation To more precisely understand the overall performance of different mixture combinations we plot the results of our grid

search with the highest compute budget in Figure 2. The results confirm that the highest performing strategies concentrate in the center of the simplex (i.e. contain a high-degree of mixing). Further, the improvement does not appear to be sensitive to the exact model mixture, as long as λ is well-enough mixed.

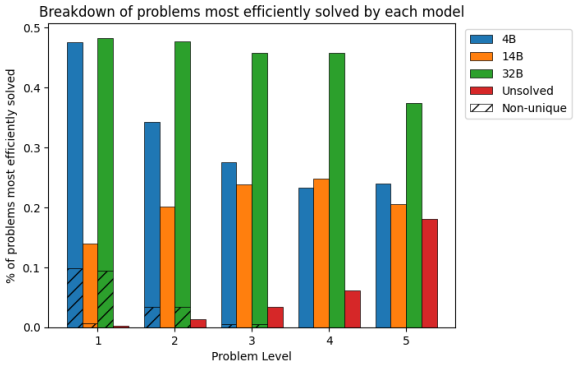

Figure 3: Breakdown of problems by most compute efficient solver across problem difficulty levels.

All models uniquely best solve a large percentage of problems Now having established the benefits of model mixing for inference-time scaling we seek to understand where these improvements come from. Via proposition 1 we know for a fixed problem Q a mixed model strategy cannot improve over the best pure strategy on Q. If the best pure strategy on all problems was always the largest 32B model, then the mixed strategy would not give any improvement at all. This suggests there is significant diversity among the best pure strategy for different types of problems.

For each problem Q in the test dataset we compute the compute normalized probability of success for each model M_1, M_2, M_3. The model with the largest cnp for Q will then be the best pure strategy for solving Q. We plot the percentage of problems best solved by each model in Figure 3 broken down by difficulty level. We find several interesting trends. At all difficulty levels all models uniquely best solve more than 15% of problems. The 4B model best solves the same percentage of level 1 problems as 32B. This percentage steadily decreases across problem difficulties, reaching 25% for the 4B model on the hardest set of problems. Surprisingly, the 14B best solves a relatively small percentage of level 1 problems compared to both other models across all difficulties. This explains why the 4B+14B model mixtures perform better than 14+32B mixtures in Figure 1. However, the subset of problems 14B best solves does increase in percentage among harder subsets, closely matching the 4B's percentage. The 32B model consistently best solves the highest percentage of problems across all levels. Yet surprisingly, the difference in percentage between 4B and and 32B does not appear to change much on the harder subsets. Remarkably, on the hardest set of problems, all models best solve between 20-40% of problems, with the rest remaining unsolved with the given compute budget. Note: the raw and compute normalized probabilities for each model across all problems are plotted in Figure 4. Figure 5 similarly plots the difference in compute normalized probability between sets of paired models.

4 Conclusion and Future Work

In this work we conducted an analysis of inference scaling laws in the mixed compute regime where samples from multiple models can be pooled given a fixed compute budget $C > 0$. We found mixing solutions from different models improved both overall performance in the most expensive compute budget regime and exhibited better scaling trends than single model strategies. As part of our analysis, we showed this is due to the complementary problem solving abilities of differently sized models. Future work might conduct more investigations into the type of data produced by mixtures of models versus single models, focusing in particular on the diversity of generated data both at the final answer and trajectory level.

Ethics Statement As with any work studying generative models, we note generative modeling can suffer from pre-existing biases in the training data. This behavior may help propagate existing societal biases present today.

Reproducibility Statement This work utilizes only open-source models and datasets making it 100% reproducible.

REFERENCES

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024. URL https://arxiv.org/abs/2407.21787.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi: 10.1109/TCIAIG.2012.2186810.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Alex Havrilla, Andrew Dai, Laura O'Mahony, Koen Oostermeijer, Vera Zisler, Alon Albalak, Fabrizio Milo, Sharath Chandra Raparthy, Kanishk Gandhi, Baber Abbasi, Duy Phung, Maia Iyer, Dakota Mahan, Chase Blagden, Srishti Gureja, Mohammed Hamdy, Wen-Ding Li, Giovanni Paolini, Pawan Sasanka Ammanamanchi, and Elliot Meyerson. Surveying the effects of quality, diversity, and complexity in synthetic data from large language models, 2024. URL https://arxiv.org/abs/2412.02980.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *NeurIPS*, 2021.

OpenAI. Learning to reason with llms. Blog post, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

Itai Shufaro, Nadav Merlis, Nir Weinberger, and Shie Mannor. On bits and bandits: Quantifying the regret-information trade-off, 2024. URL https://arxiv.org/abs/2405.16581.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models, 2024. URL https://arxiv.org/abs/2408.00724.

Kexun Zhang, Shang Zhou, Danqing Wang, William Yang Wang, and Lei Li. Scaling llm inference with optimized sample compute allocation, 2024. URL https://arxiv.org/abs/2410.22480.

A SUCCESS PROBABILITY DISTRIBUTIONS

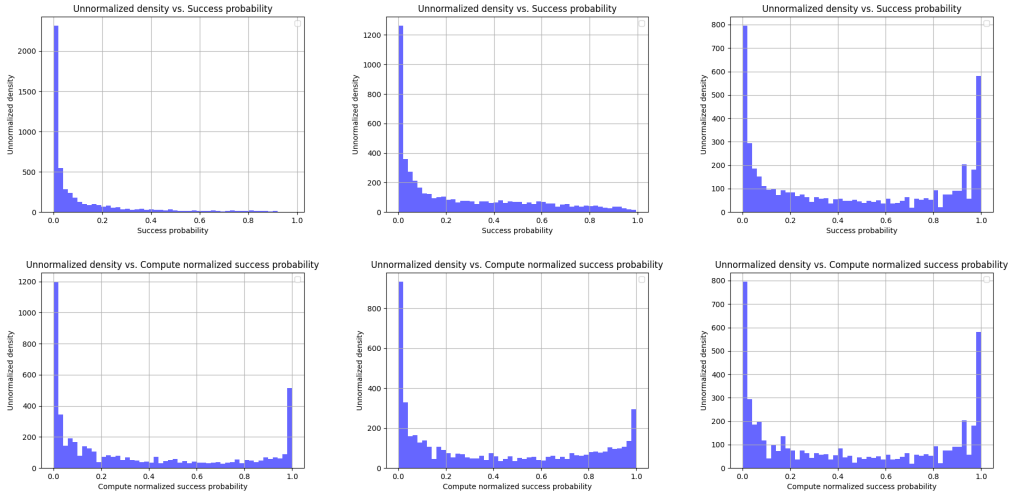

Figure 4: Distributions of raw and compute normalized success probability. The success probability of a model for a question Q is the probability of correctly solving Q.

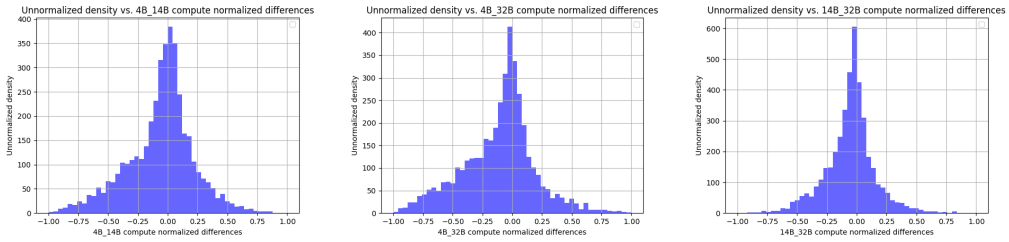

Figure 5: Distributions of the difference between compute normalized success probabilities between various models. Note: negative values indicate the larer model has a larger cnp.

B PROOF OF PROP 1

Here we present the proof of proposition 1.

Proof. First we compute $\mathbb{P}(A|Q, \mathbf{M}_\lambda)$. The probability of $\mathbf{M}_\lambda$ solving Q is equal to one minus the probability of no model in the ensemble ever solving Q, i.e.

$$\mathbb{P}(A|Q, \mathbf{M}_\lambda) = 1 - \prod_{i=1}^{k} (1 - p_i)^{\frac{\lambda_i C}{c_i}}$$

where we allocate $\lambda_i C$ compute to M_i allowing for $\frac{\lambda_i C}{c_i}$ samples. Then

$$\begin{aligned}\arg\max_{\lambda \in \Delta(k)} \mathbb{P}(A|Q, \mathbf{M}_\lambda) &= \arg\max_{\lambda \in \Delta(k)} 1 - \prod_{i=1}^{k} (1 - p_i)^{\frac{\lambda_i C}{c_i}} = \arg\min_{\lambda \in \Delta(k)} \prod_{i=1}^{k} (1 - p_i)^{\frac{\lambda_i C}{c_i}} \\ &= \arg\min_{\lambda \in \Delta(k)} \sum_{i=1}^{k} \frac{\lambda_i C}{c_i} \log((1 - p_i)) = \arg\min_{\lambda \in \Delta(k)} \sum_{i=1}^{k} \lambda_i r_i\end{aligned}$$

where we use the monotonicity of $\log$ on the second line. The final term is simply a convex combination of fixed real numbers r_i which is minimized by choosing $\lambda_i = 1$ for the smallest r_i and $\lambda_j = 0$ for $j \neq i$. □