SHAPLEY-BASED DATA VALUATION FOR LLM ALIGNMENT VIA SEQUENTIAL PREFERENCE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Data valuation is a natural framework for understanding which data sources matter most when aligning a Large Language Model (LLM) from multiple sources. The standard game-theoretic approach treats each source, or equivalently each preference dataset, as a player in a cooperative game and assigns it a contribution score through the Shapley value. In practice, however, Shapley-based valuation is computationally prohibitive because it requires aligning a separate model for every possible coalition of sources, i.e., an exponential number of alignments. We address this challenge for Direct Alignment Algorithms (DAAs), including IPO, which learn through log-policy ratios with respect to a reference policy. We show that, when a model is aligned sequentially source by source, exact optimization makes each stage contribute additively to the log-probability of a full response, up to a prompt-dependent normalization constant. This allows the log-probability assigned by any coalition to a fixed response to be reconstructed from the base policy and the policies trained on each source individually. This reduces the alignment cost of Shapley-based valuation from exponential to linear, since only one model per source needs to be trained to evaluate coalition scores. We test whether this theoretical property remains approximately valid under finite training across several base models and real-world data sources. We finally compute the Shapley values of these sources under multiple reward models, showing how their estimated contributions vary across evaluation criteria.

1 INTRODUCTION

Large language models (LLMs) are aligned using preference data drawn from multiple heterogeneous sources. These sources may differ in the prompts they cover or the procedures used to construct them, but they are ultimately combined to shape the aligned model's behavior. Preference-based alignment methods, including Reinforcement Learning from Human Feedback (RLHF) [8, 49, 20] and direct preference-optimization methods such as Direct Preference Optimization (DPO) [32] and Identity Preference Optimization (IPO) [3], make such data a central component of the alignment pipeline. This raises a natural *data valuation* question: *how should we attribute the contribution of each preference-data source to the behavior of the aligned model?* Answering this question is relevant for data markets (*what is a fair price for a dataset?*), incentive design (*how should contributors be rewarded?*), and responsibility attribution (*which sources are responsible for desirable or undesirable model behavior?*) [2, 7, 24, 36, 13, 21, 22]

Cooperative game theory, and in particular the Shapley value [35], provides a formalism for such data valuation problems by treating each data source as a player, assigning a utility to every coalition of players, and splitting the resulting utility among sources. However, directly instantiating this paradigm for LLM alignment is computationally prohibitive. The utility must be evaluated for every coalition of sources, and the number of coalitions grows exponentially with the number of sources [35, 13]. Even in classical supervised learning, this already leads to an impractical number of model retrainings. For LLM alignment, each coalition evaluation may require a full preference-optimization run, making the standard Shapley procedure infeasible at the scale of modern alignment pipelines. Emulation or distillation techniques can reduce the cost of each run [28], but they do not remove the exponential number of coalition-specific alignments.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this paper, we address this obstacle for Direct Alignment Algorithms (DAAs) [33], a broad family of preference-based alignment objectives that includes DPO [32], IPO [3], and SLiC-style losses [48]. Because these methods learn sequence-level log-policy ratios relative to a reference policy, aligning a model sequentially, one source at a time, exposes an additive structure. More precisely, under exact optimization at each stage, each source contributes a source-specific sequence-level log-ratio increment, and these increments add across updates. As a result, the score associated with any coalition can be reconstructed from the base model and the singleton models trained on individual sources, rather than by training a separate model for that coalition. This reduces the number of required alignments from exponential to linear in the number of sources. More generally, this reconstruction result is not specific to the Shapley value computation. It can be used whenever a downstream analysis requires complete-response scores for multiple coalitions of data sources.

However, this exact reconstruction statement is idealized. In practice, neural training uses finite compute and a limited optimization budget[1], so the alignment procedure may not reach the theoretical optimum at every stage. Whether coalition reconstruction remains accurate under these practical conditions is therefore an empirical question. We study this question directly across several base models and real-world preference sources. Specifically, we test whether sequential alignment is robust to the order in which sources are processed and whether coalition scores reconstructed from singleton models agree with those induced by policies trained sequentially on the same coalitions. In our main experiments, we instantiate the direct-alignment objective with IPO. We report complementary DPO results in Appendix D. We then compute Shapley values for these same sources under multiple reward models, showing how source valuations vary across evaluation criteria.

We summarize our contributions as follows.

(a) We propose a sequential preference optimization procedure for DAAs.

(b) We show that, in the idealized exact-optimization regime, the updates induced by sequential preference optimization compose additively in sequence-level log-policy-ratio space. Consequently, the log-probability assigned by any coalition policy to a fixed response can be reconstructed from the base policy and the policies trained on each source individually via a simple additive identity. This reduces the number of alignments required for Shapley-based valuation from exponential to linear in the number of sources.

(c) We study whether this idealized reconstruction remains reliable under finite neural training. Across several base models and real-world preference sources, we test whether sequential alignment is robust to the order in which sources are processed and whether coalition scores reconstructed from singleton models agree with those induced by policies trained sequentially on the same coalitions. Our main experiments instantiate the direct-alignment objective with IPO, and we report complementary DPO results in Appendix D.

(d) We apply the resulting reconstruction procedure to real-world preference sources and compute Shapley values for these same sources under multiple reward models, revealing how differently they contribute across evaluation criteria.

Related work. Two streams of Shapley value applications for aligned LLMs exist. One is focused on *model explainability* for token- or feature-level attribution of predictions [41, 43, 17, 46], and another is focused on *data valuation* for quantifying training data contributions [34, 16, 39, 29]. Our work is orthogonal to the first stream and contributes to the second. Its novelty, compared to prior data valuation studies for LLMs, is that it exploits the mathematical structure of preference-based alignment methods to reduce the computational cost of approximating coalition utilities for Shapley-based source valuation.

The computational cost of Shapley valuation has also motivated generic approximation methods that reduce the number of coalition utilities that must be evaluated, for instance through permutation Monte Carlo or other sampling schemes [6, 27, 13, 22]. These methods reduce the number of coalitions considered, but in our setting each selected coalition would still require a separate alignment run. Our method targets this complementary source of cost by reducing alignment to n trainings, regardless of the number of coalition utilities subsequently evaluated. Sampling-based Shapley approximations can therefore be combined with our reconstruction to further reduce the number of utility evaluations.

[1] Throughout the paper, we use the terms *finite training/finite optimization* to refer to this practical regime.

Our work is also related to methods that update the reference policy during offline preference optimization. Trust-region variants such as TR-DPO, TR-IPO, and TR-KTO update the reference policy within training on a single pooled dataset in order to stabilize optimization and mitigate overoptimization [14]. We also use a changing reference policy, but for a different purpose. Rather than improving a single pooled training run, we train sequentially across data sources in order to expose an additive structure in sequence-level log-policy-ratio space. We defer an extended discussion of related work to Appendix A.

2 PRELIMINARIES

Setup. Let $\mathcal{X}$ be a prompt space and let $\mathcal{Y}$ be a finite set of complete responses. The finiteness assumption can be understood as fixing a finite vocabulary and a maximum response length. We represent a language model by a policy π assigning a distribution $\pi(\cdot \mid x)$ over $\mathcal{Y}$ to every prompt $x \in \mathcal{X}$. We restrict attention to full-support policies and denote the base policy by π_0, with $\pi_0(y \mid x) > 0$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, so that the log-policy ratios used in the paper are well defined.

We consider n sources of preference data, indexed by $\mathcal{N} = \{1, \dots, n\}$. Source i is associated with a population distribution P_i over preference triples (x, y^+, y^-), where y^+ is preferred to y^- for prompt x, and with a finite training sample $\mathcal{D}_i = \{(x_m, y_m^+, y_m^-)\}_{m=1}^{M_i}$. A coalition is a subset $S \subseteq \mathcal{N}$. We write $\mathcal{D}_S = \bigcup_{i \in S} \mathcal{D}_i$ for the data provided by the sources in S.

Shapley-based data valuation. Our goal is to quantify how much each preference source $i \in \mathcal{N}$ contributes to a chosen notion of alignment quality. A standard approach is Shapley-based data valuation [13], which treats each data source as a player in a cooperative game and assigns it a value based on its average marginal contribution across coalitions. To define such a game, we first associate each coalition $S \subseteq \mathcal{N}$ with a scalar utility $v(S)$ measuring the alignment quality obtained when the alignment procedure uses the sources in S. The empty coalition corresponds to the base model, so $v(\emptyset)$ measures the utility before preference alignment.

The definition of v depends on the behavior one wishes to value. For example, a coalition may be evaluated by the average score of a reward model on a fixed set of prompts, or by the accuracy with which its complete-response scores rank held-out preferred responses above rejected responses. More generally, $v(S)$ is the scalar returned by a fixed evaluation protocol applied to the policy or response scores associated with coalition S. The same protocol is used for every coalition. Consequently, the resulting valuation is relative to the chosen utility rather than an intrinsic property of the data source itself [38, 11].

The Shapley value accounts for interactions between sources by averaging the marginal contribution of a source over all coalitions to which it can be added. For source $i \in \mathcal{N}$,

$$\phi_i(v) = \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!\,(n - |S| - 1)!}{n!} \left[v(S \cup \{i\}) - v(S) \right]. \tag{1}$$

One reason for the widespread use of the Shapley value in data valuation is that it satisfies a number of desirable axioms [35]. Among them, the efficiency axiom requires that $\sum_{i \in \mathcal{N}} \phi_i(v) = v(\mathcal{N}) - v(\emptyset)$, meaning that the total change in utility obtained by using all preference sources, relative to the base model, is fully attributed among the sources.

However, directly implementing the Shapley value is prohibitively expensive for LLM alignment. Evaluating $v(S)$ would normally require training the model associated with each nonempty coalition and then applying the evaluation protocol. This requires $2^n - 1$ alignment runs. Section 3 shows how to avoid these coalition-specific trainings while retaining the quantities needed to evaluate Equation (1).

Direct alignment algorithms. In this paper, we use a Direct Alignment Algorithm (DAA) as the alignment procedure underlying the coalition utilities $v(S)$. A DAA optimizes a policy from preference comparisons through a loss applied to differences of complete-response log-policy ratios with respect to a reference policy. Formally, let π_0 be a reference policy satisfying $\pi_0(y \mid x) > 0$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and define $h_{\pi,\pi_0}(x, y) = \log \frac{\pi(y|x)}{\pi_0(y|x)}$. Given a convex loss g and a scale

parameter $\beta > 0$, the population DAA objective associated with source i is

$$\mathcal{L}_i(\pi; \pi_0) = \mathbb{E}_{(x,y^+,y^-)\sim P_i}\left[g\left(\beta\left[h_{\pi,\pi_0}(x, y^+) - h_{\pi,\pi_0}(x, y^-)\right]\right)\right], \tag{2}$$

where $\beta > 0$ controls the scale of the reference-relative preference margin and $g : \mathbb{R} \to \mathbb{R}$ is a convex loss. DPO corresponds to $g(t) = -\log \sigma(t)$ [32], IPO to $g(t) = (t - \frac{1}{2})^2$ up to a positive multiplicative constant [3], and SLiC-style objectives correspond to hinge-type losses [48]. In practice, we replace the expectation in Equation (2) with its empirical average over $\mathcal{D}_i$.

The structure of Equation (2) will be central in Section 3, where we show how to exploit it to evaluate coalitions of preference sources without training a separate model for each coalition.

3 Efficient Shapley-based data valuation for LLM alignment

The computational bottleneck of Shapley-based data valuation is that it requires evaluating the utility of every coalition of data sources. If each coalition requires a separately aligned model, this amounts to $2^n - 1$ alignment runs for n sources. We show that, for DAAs and under exact population optimization, when alignment is performed sequentially over data sources, the information needed to evaluate any coalition can instead be recovered after training only one model per source. The proofs of all theoretical results in this section are given in Appendix B.

3.1 Sequential preference optimization

Let $S \subseteq \mathcal{N}$ be a coalition and let $\sigma = (\sigma_1, \ldots, \sigma_m)$ be an ordering of its $m = |S|$ sources. We propose a sequential preference optimization procedure that processes these sources one by one. Starting from $\pi^{(0)} = \pi_0$, stage k optimizes the DAA objective (2) associated with source σ_k, using the current policy $\pi^{(k-1)}$ as reference:

$$\pi^{(k)} \in \arg\min_{\pi} \mathcal{L}_{\sigma_k}\left(\pi; \pi^{(k-1)}\right), \qquad k = 1, \ldots, m. \tag{3}$$

In practice, Equation (3) is optimized on the finite dataset $\mathcal{D}_{\sigma_k}$. The theoretical analysis below instead considers the corresponding population objectives. For a singleton coalition $S = \{i\}$, the procedure consists of a single alignment on source i starting from the base policy π_0. We denote its exact population counterpart by $\pi_i^\star$.

3.2 Relative log-probability changes and order-independent alignment

The only additional ingredient introduced by the sequential preference optimization procedure is that the reference policy changes from one source to the next. A priori, this could make the effect of processing a source depend on where that source appears in the sequence. We therefore first ask how the update induced by a source depends on the current reference policy.

The structure of the DAA objective makes this dependence particularly simple. Recall from Equation (2) that, for a reference policy π_0, the candidate policy π enters the objective through the complete-response log-policy ratio $h_{\pi,\pi_0}(x, y) = \log \frac{\pi(y|x)}{\pi_0(y|x)}$. It is therefore convenient to reparameterize the optimization problem using the sequence-level log-policy ratio as a score coordinate, $f(x, y) = h_{\pi,\pi_0}(x, y)$. In these coordinates, the population objective associated with source i becomes

$$\mathbb{E}_{(x,y^+,y^-)\sim P_i}\left[g\left(\beta\left[f(x, y^+) - f(x, y^-)\right]\right)\right]. \tag{4}$$

Equation (4) makes explicit that the objective depends on f only through differences between scores assigned to responses to the same prompt. It is therefore unchanged when an arbitrary prompt-dependent constant $c(x)$ is added to $f(x, \cdot)$.

We formalize the ideal regime in which these score differences are well-defined.

Assumption 3.1 (Exact and identifiable population optimum)**.** *For every source i, the population objective (4) attains a finite minimum, and its minimizers are unique up to a prompt-dependent additive constant. Formally, for any two minimizers f and $\widetilde{f}$, there exists $c : \mathcal{X} \to \mathbb{R}$ such that $\widetilde{f}(x, y) = f(x, y) + c(x)$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. At every stage of the theoretical sequential procedure, the corresponding population minimum is attained.*

This assumption idealizes the optimization regime. Section 4 assesses how well the analysis developed below in this idealized regime applies to finite neural training.

Consider now source $i \in \mathcal{N}$ and an arbitrary reference policy π satisfying $\pi(y \mid x) > 0$ for all (x, y) and define

$$T_i^\star(\pi)(y \mid x) = \frac{\pi(y \mid x) \exp(f_i^\star(x, y))}{\sum_{y' \in \mathcal{Y}} \pi(y' \mid x) \exp(f_i^\star(x, y'))}.$$

Lemma 3.1 (Reference independence of relative log-probability changes)**.** *Under Assumption 3.1, $T_i^\star(\pi)$ is an exact population optimizer for source i with reference π. Moreover, for any prompt x and any two responses $y, y' \in \mathcal{Y}$,*

$$\log \frac{T_i^\star(\pi)(y \mid x)}{\pi(y \mid x)} - \log \frac{T_i^\star(\pi)(y' \mid x)}{\pi(y' \mid x)} = f_i^\star(x, y) - f_i^\star(x, y'). \tag{5}$$

The right-hand side depends only on source i and is the same for every full-support reference policy π. Hence, the change induced by source i in the relative log-probability of two responses does not depend on the reference policy.

Thus, under exact population optimization, training on source i changes the relative log-probability of any two responses by an amount determined only by that source, independently of the reference policy from which optimization starts. This immediately yields order-invariant composition.

Proposition 3.1 (Order-invariant composition across sources)**.** *Let $S \subseteq \mathcal{N}$ and let σ be any ordering of the sources in S. Under Assumption 3.1, the sequential preference optimization procedure (3) returns*

$$\pi_S^\star(y \mid x) = \frac{\pi_0(y \mid x) \exp\big(\sum_{i \in S} f_i^\star(x, y)\big)}{\sum_{y' \in \mathcal{Y}} \pi_0(y' \mid x) \exp\big(\sum_{i \in S} f_i^\star(x, y')\big)}.$$

Consequently, the resulting policy depends only on the coalition S and not on the ordering σ.

3.3 Reconstruction from singleton models

Proposition 3.1 characterizes the coalition policy through the source scores $f_i^\star$. Crucially, these scores do not need to be estimated separately. They are already encoded, up to prompt-dependent constants, in the policies obtained by aligning the base model on a single source. We refer to these as *singleton policies*. Let $\pi_i^\star = T_i^\star(\pi_0)$ be the exact policy obtained by optimizing on source i alone from the base policy. From Lemma 3.1,

$$\log \pi_i^\star(y \mid x) - \log \pi_0(y \mid x) = f_i^\star(x, y) - c_i(x),$$

where $c_i(x)$ does not depend on the response y.

For any coalition $S \subseteq \mathcal{N}$, define

$$\begin{aligned} \widehat{s}_S^\star(x, y) &= \log \pi_0(y \mid x) + \sum_{i \in S} [\log \pi_i^\star(y \mid x) - \log \pi_0(y \mid x)] \\ &= \sum_{i \in S} \log \pi_i^\star(y \mid x) + (1 - |S|) \log \pi_0(y \mid x). \end{aligned}$$

For $S = \emptyset$, this definition gives $\widehat{s}_\emptyset^\star(x, y) = \log \pi_0(y \mid x)$.

Proposition 3.2 (Exact reconstruction from singleton models)**.** *Under Assumption 3.1, for every coalition $S \subseteq \mathcal{N}$ there exists a function $c_S : \mathcal{X} \to \mathbb{R}$ such that*

$$\widehat{s}_S^\star(x, y) = \log \pi_S^\star(y \mid x) + c_S(x).$$

Consequently, for any two responses y, y' to the same prompt x,

$$\log \pi_S^\star(y \mid x) - \log \pi_S^\star(y' \mid x) = \widehat{s}_S^\star(x, y) - \widehat{s}_S^\star(x, y').$$

Proposition 3.2 is the central result of this paper. It says that the singleton policies recover any coalition log-policy up to a constant that depends on the prompt but not on the candidate response. This unidentified constant disappears whenever responses are compared or normalized within the same prompt. In particular, for any fixed nonempty candidate set $\mathcal{C}(x) \subseteq \mathcal{Y}$ and any $y \in \mathcal{C}(x)$,

$$\frac{\exp(\widehat{s}_S^\star(x, y))}{\sum_{y' \in \mathcal{C}(x)} \exp(\widehat{s}_S^\star(x, y'))} = \frac{\pi_S^\star(y \mid x)}{\sum_{y' \in \mathcal{C}(x)} \pi_S^\star(y' \mid x)}.$$

Hence, the reconstruction exactly preserves pairwise score differences, rankings, argmax decisions, and normalized distributions over any fixed candidate set.

3.4 SHAPLEY-BASED VALUATION WITH n ALIGNMENTS

We can now return to the data valuation problem introduced in Section 2. Let $\mathcal{V} : \mathbb{R}^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}$ denote the fixed evaluation protocol used to define coalition utility. We require only that $\mathcal{V}$ be invariant to adding a prompt-dependent constant to all response scores for that prompt, i.e.,

$$\mathcal{V}(s + c) = \mathcal{V}(s) \tag{6}$$

for any function $c(\cdot)$ that does not depend on the response.

In the exact population regime, we define the coalition utility by $v(S) = \mathcal{V}(\log \pi^\star_S)$. Proposition 3.2 and Equation (6) imply $v(S) = \mathcal{V}(\widehat{s}^\star_S)$. Thus, every coalition utility can be recovered from the base policy and the singleton policies, without training a model on that coalition.

The corresponding Shapley value of source i is therefore

$$\phi_i(v) = \sum_{S\subseteq\mathcal{N}\setminus\{i\}} \frac{|S|!\,(n-|S|-1)!}{n!} \left[\mathcal{V}\left(\widehat{s}^\star_{S\cup\{i\}}\right) - \mathcal{V}(\widehat{s}^\star_S)\right].$$

The computational cost of Shapley valuation has two distinct components: training the models associated with coalitions and evaluating their utilities. Our method reduces the first, and typically much more expensive, component. A direct implementation requires training one aligned model for each of the $2^n - 1$ nonempty coalitions, whereas our reconstruction requires only the n singleton policies $\pi^\star_1, \ldots, \pi^\star_n$. Exact Shapley computation still requires evaluating the utilities of all 2^n coalitions, but these utilities can now be obtained from reconstructed coalition scores without any additional alignment. Our method can therefore be combined with standard Shapley approximation techniques, such as Monte Carlo permutation sampling, to also reduce the number of coalition utilities that must be evaluated when exhaustive enumeration is too costly [6, 27, 13].

In practice, the singleton policies are obtained by finite neural optimization rather than exact population optimization. Given the learned singleton models $\pi_1, \ldots, \pi_n$, we therefore use

$$\widehat{s}_S(x, y) = \log \pi_0(y \mid x) + \sum_{i\in S} [\log \pi_i(y \mid x) - \log \pi_0(y \mid x)] \tag{7}$$

and define the reconstructed coalition utility $\widehat{v}(S) = \mathcal{V}(\widehat{s}_S)$.

Unlike under exact population optimization, Equation (7) is not guaranteed to match the complete-response scores of a coalition obtained by sequentially training on sources in it. Section 4 therefore empirically tests whether the properties established in the idealized regime remain approximately valid in practice. Once we assess this empirical approximation, Section 5 instantiates $\mathcal{V}$ with alignment criteria and uses $\widehat{v}$ to compute Shapley values for real-world data sources.

4 FINITE-TRAINING VALIDATION

Section 3 establishes two consequences of sequential preference alignment under exact population optimization that are central to our method. First, sequential optimization over a coalition is independent of the order in which its sources are processed. Second, the behavior of any coalition can be reconstructed at the level of complete-response scores using only the policies trained on individual sources. However, neither property is guaranteed to hold exactly after finite neural-network training. We therefore test these two consequences directly.

4.1 EXPERIMENTAL PROTOCOL

We construct $n = 9$ preference sources from the UltraFeedback dataset [9] by partitioning prompts according to their original data provenance. For each source i, the dataset $\mathcal{D}_i$ contains preference triples (x, y^+, y^-) from that provenance, where y^+ and y^- are the highest- and lowest-scoring valid completions available for prompt x, respectively.

We evaluate finite-training behavior across SmolLM [5], Qwen2.5 [31], Llama3.2 [15] and Phi-3.5 [1] models. All models are aligned with IPO [3]. Full experimental details are provided in Appendix C.

We consider coalitions of sizes $|S| \in \{3, 6, 9\}$. For each evaluated coalition, we specify three source orderings $\Sigma_S = \{\sigma_1, \sigma_2, \sigma_3\}$. For an ordering $\sigma \in \Sigma_S$, let $\pi_{S,\sigma}$ denote the policy obtained by the sequential preference optimization procedure specified in Section 3.1. We train different orders under the same seed and identical experimental conditions. We also repeat the same ordered training with independent optimization seeds to estimate variability due to finite training. Appendix C lists the exact coalitions, source orders, and repetitions.

All comparisons are performed on the same held-out evaluation set $\mathcal{E} = \{(x_j, y_j^+, y_j^-)\}_{j=1}^M$.

4.2 A COMMON DISCREPANCY MEASURE

Proposition 3.2 shows that $\widehat{s}_S^\star(x, y)$ recovers $\log \pi_S^\star(y \mid x)$ up to a prompt-dependent additive constant. Because this constant is the same for all responses to a given prompt, it cancels when taking the difference between the scores assigned to two responses. We therefore evaluate finite-training deviations through such pairwise differences.

For a policy π, define its preference gap on evaluation example $(x_j, y_j^+, y_j^-) \in \mathcal{E}$ as $G_\pi^{(j)} = \log \pi(y_j^+ \mid x_j) - \log \pi(y_j^- \mid x_j)$. Given two vectors of preference gaps G and G', we measure their discrepancy by

$$d(G, G') = \frac{1}{M} \sum_{j=1}^{M} \left| G^{(j)} - G'^{(j)} \right|.$$

However, raw values of d are not directly comparable across models and coalition sizes, since the scale of finite-training variability itself may differ. To obtain a comparable reference scale, for each considered coalition S, we fix one reference ordering σ_{ref}. We train this same ordered coalition R times, using different optimization seeds $r \in [R]$, while keeping the data, source order, and all other training choices fixed. Let $\pi_{S,\sigma_{\text{ref}}}^{(r)}$ denote the policy obtained in run r. We define the *retraining floor*[2] as the average pairwise discrepancy between these R runs, i.e.,

$$d_{\text{floor}}(S) = \frac{2}{R(R-1)} \sum_{1 \le r < r' \le R} d\left(G_{\pi_{S,\sigma_{\text{ref}}}^{(r)}}, G_{\pi_{S,\sigma_{\text{ref}}}^{(r')}} \right).$$

Both experiments below report their discrepancy relative to this same quantity.

4.3 ROBUSTNESS TO SOURCE ORDER

Proposition 3.1 predicts that, under exact population optimization, the final coalition policy does not depend on the source ordering. Under finite training, we measure how far this order invariance degrades. For two distinct orderings $\sigma, \tau \in \Sigma_S$, we define the order discrepancy as $d_{\text{order}}(S; \sigma, \tau) = d\big(G_{\pi_{S,\sigma}}, G_{\pi_{S,\tau}}\big)$, and normalize it by the retraining floor to obtain

$$\rho_{\text{order}}(S; \sigma, \tau) = \frac{d_{\text{order}}(S; \sigma, \tau)}{d_{\text{floor}}(S)}.$$

$\rho_{\text{order}} = 1$ means that changing the source order causes a discrepancy equal to ordinary retraining variability.

4.4 RECONSTRUCTION FIDELITY

We next test Proposition 3.2 under finite training. Equation (7) assigns a reconstructed score $\widehat{s}_S(x, y)$ to each response. We define the corresponding reconstructed preference gap on evaluation example $(x_j, y_j^+, y_j^-) \in \mathcal{E}$ as $\widehat{G}_S^{(j)} = \widehat{s}_S(x_j, y_j^+) - \widehat{s}_S(x_j, y_j^-)$.

For each ordering σ, we compare this reconstructed gap with the gap of the corresponding sequentially trained policy $d_{\text{recon}}(S; \sigma) = d\left(\widehat{G}_S, G_{\pi_{S,\sigma}}\right)$. We again normalize by the same retraining

[2] We use *floor* as shorthand for this empirical reference scale. It is not assumed to be a theoretical lower bound.

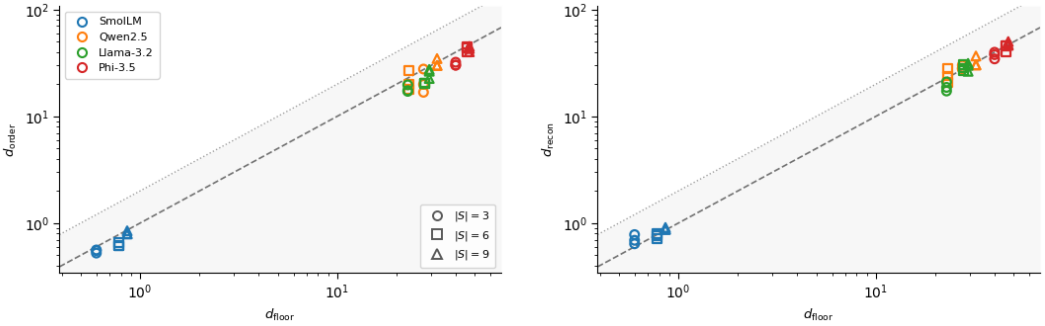
floor,

$$\rho_{\text{recon}}(S;\sigma) = \frac{d_{\text{recon}}(S;\sigma)}{d_{\text{floor}}(S)}.$$

$\rho_{\text{recon}} = 1$ means that the reconstruction discrepancy is equal to the retraining variability measured under the fixed reference ordering.

4.5 Results

Figure 1 summarizes both tests using the same representation.



(a) Order robustness

(b) Reconstruction fidelity

Figure 1: Finite-training validation of the two properties underlying efficient Shapley-based valuation for LLM alignment. Each point compares a measured discrepancy (vertical axis) with the retraining floor for the corresponding model and coalition (horizontal axis). Left: d_{order}, the discrepancy between sequential trainings that differ in source order. Right: d_{recon}, the discrepancy between reconstructed coalition gaps and an explicitly trained sequential coalition. The dashed diagonal corresponds to $\rho = 1$. The shaded region indicates $y \leq 2x$, i.e., discrepancies no larger than twice the retraining floor. Colors identify the four base models and marker shapes indicate coalition size $|S| \in \{3, 6, 9\}$. Three prespecified source orderings are evaluated for each coalition size.

Across all evaluated models, coalition sizes, and source orderings, both order and reconstruction discrepancies remain within twice the retraining floor. Thus, in all tested configurations, changing the source order or reconstructing a coalition from singleton policies introduces deviations that remain on the same scale as the variability induced by finite training itself. These results do not imply exact order invariance or exact reconstruction under finite optimization, but they show that the departures from the ideal theory remain controlled relative to retraining variability.

5 Application to Shapley-based Data Valuation

We finally apply the reconstruction method to estimate the contribution of the $n = 9$ preference sources. Since Section 4 showed that finite-training errors remain comparable to ordinary retraining variability, we use the learned singleton policies to evaluate all coalitions without additional alignment.

Let $\mathcal{E} = \{(x_j, y_j^+, y_j^-)\}_{j=1}^M$ be the held-out evaluation set introduced in Section 4, and let $\mathcal{X}_\mathcal{E} = \{x_j\}_{j=1}^M$ denote its prompts. For each $x \in \mathcal{X}_\mathcal{E}$, we generate once a fixed candidate set $\mathcal{C}(x) = \{y_1, \ldots, y_K\}$ from a coalition-independent proposal policy. We use the same candidates for every coalition. For $S \subseteq \mathcal{N}$, we reconstruct their scores using Equation (7) and select

$$\widehat{y}_S(x) = \arg\max_{y \in \mathcal{C}(x)} \widehat{s}_S(x, y). \tag{8}$$

Under exact population optimization, Proposition 3.2 implies that $\widehat{s}_S^\star(x, y)$ differs from $\log \pi_S^\star(y \mid x)$ only by a prompt-dependent constant. Hence, Equation (8) selects the same response as the exact coalition policy within the fixed candidate set.

We evaluate the selected responses with three reward models, namely Skywork-Reward-V2 [26], GRM [45], and UltraRM [9], respectively denoted as $\mathcal{R} = \{r_1, r_2, r_3\}$ where each $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ assigns a scalar reward $r(x, y)$ to response y for prompt x. Because their numerical scales may differ and to make the comparison across reward models visually interpretable, each reward is standardized over the common set of evaluated prompt-candidate pairs, i.e., $\widetilde{r}(x, y) = \frac{r(x,y)-\mu_r}{\sigma_r}$ where μ_r and σ_r denote the mean and standard deviation of reward r over all $x \in \mathcal{X}_E$ and $y \in C(x)$. The utility of coalition S under reward r is then $\widehat{v}_r(S) = \frac{1}{|\mathcal{X}_\mathcal{E}|} \sum_{x \in \mathcal{X}_\mathcal{E}} \widetilde{r}(x, \widehat{y}_S(x))$.

For each reward r and each source i, we compute the Shapley value $\phi_i(\widehat{v}_r)$ from the reconstructed coalition utilities[3]. Figure 2 shows each source's Shapley value when the base model is Qwen2.5, under the three reward models. Equivalent results for SmolLM, Llama3.2 and Phi-3.5 (the other models evaluated in Section 4) are reported in Appendix E.

The resulting valuations are broadly consistent across reward models. Sources receiving a large positive contribution under one reward tend to remain positively valued under others, while several sources cluster near the origin and therefore contribute only marginally to the corresponding games. ShareGPT and Ultrachat are separated from the remaining sources along the r_3 axis. By contrast, most `flan`-derived sources lie closer to the origin.

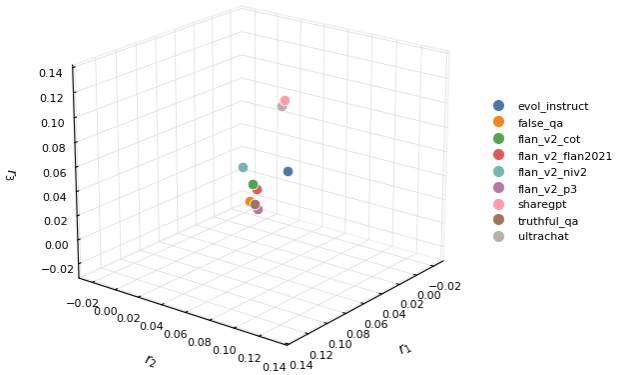


Figure 2: Shapley values of the $n = 9$ preference sources for Qwen2.5 under three standardized reward models. Each source i is represented by $(\phi_i(\widehat{v}_{r_1}), \phi_i(\widehat{v}_{r_2}), \phi_i(\widehat{v}_{r_3}))$. Along axis r_k, coordinates are expressed in standard-deviation units of the corresponding reward score.

More generally, our method makes Shapley-based valuation tractable, providing an interpretable view of which data sources contribute to the aligned model's performance and how these contributions depend on the chosen reward.

6 Conclusion

We introduced a method for Shapley-based valuation of preference data sources that avoids training a separate model for every coalition. Under exact population optimization, we showed that sequentially optimizing a DAA over the sources in a coalition gives the same final policy regardless of their order. We further showed that, for any complete response, the log-probability assigned by this coalition policy can be recovered up to a prompt-dependent constant from the base policy and the policies trained on each source individually. As a result, evaluating coalitions requires only n singleton alignments instead of $2^n - 1$ coalition-specific alignments. We then tested whether these theoretical properties persist under finite training. Across several models, changing the source order and reconstructing coalition scores from singleton policies both introduced discrepancies no larger than twice the retraining variability. Finally, we applied our method to $n = 9$ real-world preference sources and computed their Shapley values under three reward models, a computation that would otherwise require $2^9 - 1 = 511$ alignment runs and be prohibitively expensive in practice.

Limitations. Our reconstruction operates at the level of sequence log-probabilities for fixed complete responses. It does not reconstruct a generative coalition policy. Appendix F.2 explains why obtaining next-token probabilities would additionally require marginalization over future continuations. Extending the method to support direct generation is a natural direction for future work.

[3]Full experimental details are given in Appendix C.

AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing and to assist with code generation. All authors carefully reviewed the writing. AI-assisted code was reviewed and checked by at least one author. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

An anonymized code package is provided in the supplementary material. It implements the data preparation, alignment, evaluation, valuation, and plotting procedures used in the paper. Appendix C documents full experimental protocols, including datasets, hyperparameters, and compute settings, and cross-references them at the relevant points in the main text. Appendix B contains all missing proofs of theoretical results in the main text.

REFERENCES

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

[2] Anish Agarwal, Munther Dahleh, and Tuhin Sarkar. A marketplace for data: An algorithmic solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 701–726, 2019. doi: 10.1145/3328526.3329589.

[3] Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

[4] J.F. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19(2):317–343, 1965.

[5] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. SmolLM: Blazingly fast and remarkably powerful, 2024.

[6] Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009. doi: 10.1016/j.cor.2008.04.004.

[7] Yiling Chen, Stephen Chong, Ian A. Kash, Tal Moran, and Salil Vadhan. Truthful mechanisms for agents that value privacy. *ACM Transactions on Economics and Computation*, 4(3):1–30, 2016. doi: 10.1145/2892555.

[8] Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

[9] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Zhu Wei, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. UltraFeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*, 2023. URL https://arxiv.org/abs/2310.01377.

[10] Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. Controlled text generation via language model arithmetic. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=SLw9fp4yI6.

[11] Hannah Diehl and Ashia C. Wilson. Semivalue-based data valuation is arbitrary and gameable. *arXiv preprint arXiv:2506.12619*, 2025.

[12] Pradeep Dubey, Abraham Neyman, and Robert James Weber. Value theory without efficiency. *Mathematics of Operations Research*, 6(1):122–128, 1981.

[13] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2242–2251, 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

[14] Alexey Gorbatovski, Boris Shaposhnikov, Alexey Malakhov, Nikita Surnachev, Yaroslav Aksenov, Ian Maksimov, Nikita Balagansky, and Daniil Gavrilov. Learn your reference model for real good alignment. *arXiv preprint arXiv:2404.09656*, 2024.

[15] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[16] Yexiao He, Ziyao Wang, Zheyu Shen, Guoheng Sun, Yucong Dai, Yongkai Wu, Hongyi Wang, and Ang Li. SHED: Shapley-based automated dataset refinement for instruction fine-tuning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[17] Miriam Horovicz and Roni Goldshmidt. TokenSHAP: Interpreting large language models with monte carlo shapley value estimation. In *Proceedings of the 1st Workshop on NLP for Science*, pp. 1–8, 2024. doi: 10.18653/v1/2024.nlp4science-1.1. URL https://aclanthology.org/2024.nlp4science-1.1/.

[18] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

[19] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9525–9587. PMLR, 2022.

[20] Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E. Turner, and Douglas Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with KL-control. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1645–1654, 2017. URL https://proceedings.mlr.press/v70/jaques17a.html.

[21] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *Proceedings of the VLDB Endowment*, 12(11):1610–1623, 2019. doi: 10.14778/3342263.3342637.

[22] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. Towards efficient data valuation based on the shapley value. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1167–1176, 2019. URL https://proceedings.mlr.press/v89/jia19a.html.

[23] Kevin Fu Jiang, Weixin Liang, James Zou, and Yongchan Kwon. OpenDataVal: A unified benchmark for data valuation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 28624–28647, 2023.

[24] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1885–1894, 2017.

[25] Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. *arXiv preprint arXiv:2110.14049*, 2021.

[26] Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, et al. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. In *International Conference on Learning Representations*, volume 2026, pp. 133805–133838, 2026.

[27] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. Bounding the estimation error of sampling-based shapley value approximation with/without stratifying. *arXiv preprint arXiv:1306.4265*, 2013. URL https://arxiv.org/abs/1306.4265.

[28] Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D. Manning. An emulator for fine-tuning large language models using small language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=E0na7iJd2E.

[29] Hyeonseok Moon, Jaehyung Seo, Seonmin Koo, Jinsung Kim, Young-kyoung Ham, Jiwon Moon, and Heuiseok Lim. LimaCost: Data valuation for instruction tuning of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 12841–12854, 2025. doi: 10.18653/v1/2025.findings-emnlp.688. URL https://aclanthology.org/2025.findings-emnlp.688/.

[30] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930, 2020.

[31] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

[32] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

[33] Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sushil Sikchi, Joey Hejna, Brad Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. *Advances in Neural Information Processing Systems*, 37:126207–126242, 2024.

[34] Stephanie Schoch, Ritwick Mishra, and Yangfeng Ji. Data selection for fine-tuning large language models using transferred shapley values. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 266–275, 2023. doi: 10.18653/v1/2023.acl-srw.37. URL https://aclanthology.org/2023.acl-srw.37/.

[35] Lloyd S. Shapley. A value for n-person games. In *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, 1953.

[36] Rachael Hwee Ling Sim, Yehong Zhang, Mun Choon Chan, and Bryan Kian Hsiang Low. Collaborative machine learning with incentive-aware model rewards. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[37] Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning: "ingredients", strategies, and open challenges. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 5607–5614. International Joint Conferences on Artificial Intelligence Organization, 2022. doi: 10.24963/ijcai.2022/782. Survey Track.

[38] Mélissa Tamine, Benjamin Heymann, Maxime Vono, and Patrick Loiseau. On the impact of the utility in semivalue-based data valuation. In *International Conference on Learning Representations*, volume 2026, pp. 103780–103823, 2026.

[39] Jiachen Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data shapley in one training run. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HD6bWcj87Y.

[40] Jiachen T Wang and Ruoxi Jia. Data banzhaf: A robust data valuation framework for machine learning. In *International conference on artificial intelligence and statistics*, pp. 6388–6421. PMLR, 2023.

[41] Jingtan Wang, Xiaoqiang Lin, Rui Qiao, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Helpful or harmful data? fine-tuning-free shapley attribution for explaining language model predictions. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[42] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 2022.

[43] Yingtai Xiao, Yuqing Zhu, Sirat Samyoun, Wanrong Zhang, Jiachen T. Wang, and Jian Du. TokenShapley: Token-level context attribution with shapley value. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3882–3894, 2025. doi: 10.18653/v1/2025.findings-acl.200. URL https://aclanthology.org/2025.findings-acl.200/.

[44] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[45] Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for llms. *Advances in Neural Information Processing Systems*, 37:62279–62309, 2024.

[46] Zikun Ye and Hema Yoganarasimhan. Fair document valuation in llm summaries via shapley values. *arXiv preprint arXiv:2505.23842*, 2025. URL https://arxiv.org/abs/2505.23842.

[47] Chih-Kuan Yeh, Joon Sik Kim, Ian En-Hsu Yen, and Pradeep Ravikumar. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems*, volume 31, pp. 9311–9321, 2018.

[48] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

[49] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019. URL https://arxiv.org/abs/1909.08593.

Outline. The appendices are organized as follows.

- Appendix A provides an extended discussion of related work.
- Appendix B contains the proofs omitted from the main text.
- Appendix C provides experimental details for all empirical results reported in the main paper.
- Appendix D reports additional experimental results.
- Appendix E provides additional figures complementing the results presented in the main paper.
- Appendix F contains additional discussions and technical remarks.

A Extended related work

A.1 Shapley-based data valuation

Shapley-based data valuation has become popular in recent years because it provides an axiomatic framework for attributing a trained model's performance to the data used to train it. In this framework, each training example is treated as a player in a cooperative game. The utility of a coalition is the performance of a model trained on the corresponding subset, and the Shapley value [35] assigns each example its average marginal contribution over coalitions [13, 22]. This formulation has been used for data debugging, data selection, noisy-label detection, and dataset pricing [37, 23]. The same cooperative-game principle extends from data points to datasets or sources: each contributor provides a dataset, and the goal is to value each source's contribution to the final model. This source-level formulation is the one considered in our paper.

The appeal of the Shapley value comes from its axiomatic characterization. It is the unique valuation rule satisfying four key axioms:

1. *Null player.* If a player has zero marginal contribution to every coalition, i.e., $u(S \cup \{i\}) = u(S) \forall S \subseteq \mathcal{I} \setminus \{i\}$, then its value is zero: $\phi_i(u) = 0$.
2. *Symmetry.* If two players have identical marginal contributions to every coalition, i.e., $u(S \cup \{i\}) = u(S \cup \{j\}) \forall S \subseteq \mathcal{I} \setminus \{i, j\}$, then they receive the same value: $\phi_i(u) = \phi_j(u)$.
3. *Efficiency.* The total value is distributed among all players: $\sum_{i \in \mathcal{I}} \phi_i(u) = u(\mathcal{I}) - u(\emptyset)$.
4. *Additivity.* For any two games u and w on the same player set, $\phi_i(u + w) = \phi_i(u) + \phi_i(w), \forall i \in \mathcal{I}$.

A growing body of work applies Shapley-based data valuation to language model fine-tuning and instruction tuning. [34] proposes transferred Shapley values for data selection in LLM fine-tuning, using value information from simpler models to reduce valuation costs. [16] introduces SHED, a Shapley-based automated dataset refinement framework for instruction fine-tuning. [29] proposes LimaCost, a data valuation method for instruction tuning that estimates an example's value by the amount of reference data needed to approximate its gradient. These methods share our goal of identifying valuable data for LLM adaptation, but they do not exploit the algebraic structure of preference-optimization objectives to reconstruct complete-response coalition scores from singleton models.

Finally, several works study Shapley values for LLMs from an explainability rather than a training-data valuation perspective. Token-level methods such as TokenSHAP and TokenShapley attribute model outputs to prompt or context tokens [17, 43]. Other work studies document-level attribution in generated summaries using Shapley-style document valuation [46]. These methods explain a prediction, a generated answer, or a summary. Our work instead values preference datasets used during alignment: the players are data sources, and the utility of a coalition is the behavior of the aligned model obtained from those sources.

A.2 Efficient Shapley value estimation methods

The main computational obstacle in Shapley-based data valuation is the exponential number of coalitions. For n players, exact computation requires evaluating utilities for all 2^n coalitions. Classical

approximation methods, therefore, estimate the Shapley value by sampling coalitions or permutations. Permutation Monte Carlo estimates the Shapley value by averaging marginal contributions along random permutations of players [6, 27, 13]. Stratified sampling improves this approach by allocating samples across coalition sizes [27]. Group-testing methods estimate the Shapley vector via randomized coalition queries and pairwise-difference constraints [22]. These methods are generic: they reduce the number of coalitions queried, but each queried coalition must still be evaluated by running the learning procedure associated with it.

Other works exploit structure in specific learning algorithms. For nearest-neighbor classifiers, [21] derives efficient task-specific data valuation procedures that avoid generic retraining over coalitions. More recently, [39] show how Data Shapley values can be estimated from a single training run by accumulating contribution estimates along the training trajectory. These methods show that Shapley computation can become practical when one uses the training algorithm's structure rather than treating the learning process as a black-box utility oracle.

Our work follows the same high-level philosophy (using algorithmic structure to reduce valuation cost) but in a different way. We consider source-level valuation for preference-based LLM alignment. Rather than directly approximating the Shapley summation, we reduce the cost of evaluating coalition utilities. Under the exact-optimization structure of Direct Alignment Algorithms, source-level updates compose additively in sequence-level log-policy-ratio space. This makes it possible to train one singleton model per source and reconstruct the complete-response scores needed to evaluate coalition utilities, without coalition-specific alignment. Thus, our method is complementary to Monte Carlo or group-testing estimators: once coalition utilities can be cheaply queried via reconstruction, standard Shapley approximation methods can still be used to reduce the number of coalition utility evaluations.

A.3 Semivalue-based data valuation

The Shapley value is one member of the broader family of semivalues [12]. Semivalues assign value through weighted averages of marginal contributions, with coalitions of the same size receiving the same weight, while allowing the weighting across coalition sizes to differ from that of the Shapley value. In particular, they need not satisfy the efficiency property that uniquely characterizes the Shapley weighting within this family.

This additional flexibility has been studied for data valuation. Kwon & Zou [25] introduce Beta Shapley, which uses a family of cardinality-dependent weights to reduce the effect of noisy marginal contributions and contains Data Shapley as a special case. [40] study the Banzhaf value as a data valuation rule and show that its alternative coalition weighting can provide greater robustness to stochastic utility evaluations. These works show that the choice of weights across coalition sizes can be useful when properties beyond Shapley efficiency are desired.

A related line of work studies the sensitivity of semivalue-based data valuation to the definition or estimation of the underlying utility. [38] analyzes how changes in the utility function affect semivalue-based rankings, while [11] studies the extent to which such valuations can depend on an underspecified utility. These results emphasize that, for Shapley values as well as other semivalues, data values are defined relative to both a coalition utility and a choice of marginal-contribution weighting. We discuss in Appendix F.1 how our coalition reconstruction can be used with arbitrary semivalue weights.

A.4 Alternative data valuation frameworks

Beyond Shapley-based methods, and more generally semivalue-based methods, several approaches assign value or influence scores to data without using the exact cooperative-game formulation. Influence functions approximate the effect of upweighting or removing a training point through derivatives of the learned parameters [24]. TracIn and related methods estimate data influence by accumulating gradient similarities along optimization trajectories [30]. Representer-point methods decompose predictions through training examples in representation space [47]. Data models learn a predictive model of how training subsets affect model outputs, providing a flexible empirical representation of data influence [19]. These methods are often more scalable than exact Shapley

computation, but they generally define different notions of influence rather than the Shapley value itself.

In the LLM setting, data selection and data attribution have also been studied through non-Shapley criteria. Some methods select instruction-tuning examples based on diversity, quality scores, gradient similarity, or influence-style quantities. These approaches are useful for improving fine-tuning efficiency, but they typically do not preserve the cooperative-game interpretation of marginal contribution across data sources. By contrast, our objective is to keep the Shapley value as the valuation rule and make its coalition utilities tractable for preference-aligned LLMs.

A related robustness line of research examines how sensitive semivalue-based data valuations are to modeling choices. [38] shows that changing the utility function can substantially alter semivalue-based rankings and introduces tools for analyzing utility-dependent robustness. [11] argue that semivalue-based data valuation can be arbitrary or gameable when the utility is underspecified. These works highlight that data values are meaningful only relative to a clearly specified utility.

A.5 Language model arithmetic and model merging

Our reconstruction rule is closely related to recent work showing that pretrained or fine-tuned models can often be combined algebraically. Language model arithmetic composes language models at inference time by combining their output distributions or logits, enabling controlled text generation without additional fine-tuning [10]. Our method is related in that it also combines information encoded by independently aligned models, but the object being reconstructed is different. Our theoretical result is derived at the level of complete-response log-probabilities: under exact population optimization, the score assigned by a coalition policy to any fixed complete response can be recovered, up to a prompt-dependent additive constant, from the base policy and the corresponding singleton policies. In practice, we use this identity only to score and compare fixed complete responses. We do not locally combine next-token distributions to define a generative coalition policy. Appendix F discusses this distinction in detail.

Model merging and task arithmetic provide another related perspective. Model soups average weights of fine-tuned models and can improve accuracy without increasing inference cost [42]. Task arithmetic represents fine-tuning effects as task vectors in weight space and combines these vectors to edit model behavior [18]. TIES-merging studies interference between task vectors and proposes sign-resolution and trimming steps to merge models more robustly [44]. These methods demonstrate that independently fine-tuned models can sometimes be combined without retraining. However, they generally operate in parameter space and are motivated by multi-task adaptation or model editing. Our method instead operates in complete-response log-probability space and is tied specifically to preference optimization: the composition rule reconstructs the complete-response scores associated with a coalition from the base and singleton policies, enabling Shapley utilities to be evaluated without coalition-specific alignment.

Our work is also related to methods that update the reference policy during offline alignment. Trust-region variants such as TR-DPO, TR-IPO, and TR-KTO update the reference policy during training to improve stability and mitigate overoptimization [14]. Sequential Preference Optimization also updates the reference policy, but for a different purpose. Trust-region methods use reference updates within a single training run on a pooled dataset. We use source-wise reference updates to expose an additive structure across data sources. This distinction is important: our sequential procedure is not primarily an optimization trick, but a way to define the alignment procedure associated with each coalition so that, under the assumptions of our theory, its complete-response scores can be reconstructed from singleton policies and used for Shapley-based data valuation.

B MISSING PROOFS

This appendix provides complete proofs of the theoretical results stated in Section 3. Throughout, we work under Assumption 3.1 and with full-support policies, as in the main text.

B.1 PROOF OF LEMMA 3.1

Lemma B.1 (Restatement of Lemma 3.1)**.** *Fix a source $i \in \mathcal{N}$, let $f_i^\star$ be any minimizer of the population score objective (4), and let π be any full-support reference policy. Define*

$$T_i^\star(\pi)(y \mid x) = \frac{\pi(y \mid x)\exp(f_i^\star(x,y))}{Z_{i,\pi}(x)}, \qquad Z_{i,\pi}(x) := \sum_{u\in\mathcal{Y}} \pi(u \mid x)\exp(f_i^\star(x,u)). \tag{9}$$

Then, under Assumption 3.1, $T_i^\star(\pi)$ is an exact population optimizer of $\mathcal{L}_i(\cdot;\pi)$. Moreover, for every prompt x and every pair of responses $y, y' \in \mathcal{Y}$,

$$\log \frac{T_i^\star(\pi)(y \mid x)}{\pi(y \mid x)} - \log \frac{T_i^\star(\pi)(y' \mid x)}{\pi(y' \mid x)} = f_i^\star(x,y) - f_i^\star(x,y'). \tag{10}$$

Therefore, the induced change in relative log-probability depends on the source i but not on the choice of reference policy π.

Proof. For convenience, write the population objective in score coordinates as

$$\mathcal{R}_i(f) := \mathbb{E}_{(x,y^+,y^-)\sim P_i}\left[g\big(\beta\left[f(x,y^+) - f(x,y^-)\right]\big)\right].$$

By Assumption 3.1, $f_i^\star$ attains its minimum.

Because $\mathcal{Y}$ is finite and π has full support, $Z_{i,\pi}(x)$ in (9) is finite and strictly positive. Hence $T_i^\star(\pi)$ is a well-defined full-support policy. Its log-policy ratio with respect to π is

$$\begin{aligned} h_{T_i^\star(\pi),\pi}(x,y) &= \log \frac{T_i^\star(\pi)(y \mid x)}{\pi(y \mid x)} \\ &= f_i^\star(x,y) - \log Z_{i,\pi}(x). \end{aligned} \tag{11}$$

Thus the reference policy affects the realized score through the prompt-dependent normalization term $\log Z_{i,\pi}(x)$.

Since this term is constant across responses to the same prompt, it cancels inside every preference comparison:

$$\begin{aligned} &h_{T_i^\star(\pi),\pi}(x,y^+) - h_{T_i^\star(\pi),\pi}(x,y^-) \\ &\quad = f_i^\star(x,y^+) - f_i^\star(x,y^-). \end{aligned}$$

Substituting into the DAA objective therefore gives

$$\mathcal{L}_i(T_i^\star(\pi);\pi) = \mathcal{R}_i(f_i^\star). \tag{12}$$

For any other full-support policy $\tilde{\pi}$, its log-policy ratio $h_{\tilde{\pi},\pi}$ defines a valid score function, and therefore

$$L_i(\tilde{\pi};\pi) = R_i(h_{\tilde{\pi},\pi}) \geq R_i(f_i^\star).$$

Together with (12), this proves that $T_i^\star(\pi)$ is an exact population optimizer of $\mathcal{L}_i(\cdot;\pi)$.

We now establish the reference-independence statement. For any two responses $y, y' \in \mathcal{Y}$, subtracting (11) at y' from the same identity at y gives

$$\begin{aligned} &\log \frac{T_i^\star(\pi)(y \mid x)}{\pi(y \mid x)} - \log \frac{T_i^\star(\pi)(y' \mid x)}{\pi(y' \mid x)} \\ &\quad = f_i^\star(x,y) - f_i^\star(x,y'), \end{aligned} \tag{13}$$

because $\log Z_{i,\pi}(x)$ cancels.

It remains to verify that the right-hand side is itself independent of the reference policy. The objective $\mathcal{R}_i$ in (B.1) depends on the source distribution P_i, g, and β, but not on π. Moreover, Assumption 3.1 implies that any two minimizers $f_i^\star$ and $\widetilde{f}_i^\star$ satisfy

$$\widetilde{f}_i^\star(x, y) = f_i^\star(x, y) + a(x)$$

for some $a : \mathcal{X} \to \mathbb{R}$. Hence, for any y, y',

$$\begin{aligned}\widetilde{f}_i^\star(x, y) - \widetilde{f}_i^\star(x, y') &= f_i^\star(x, y) + a(x) - f_i^\star(x, y') - a(x) \\ &= f_i^\star(x, y) - f_i^\star(x, y').\end{aligned}$$

Therefore the pairwise difference

$$\Delta_i(x; y, y') := f_i^\star(x, y) - f_i^\star(x, y')$$

is uniquely determined by source i and does not depend on the choice of minimizer.

Now let π' be any other full-support reference policy. Repeating (13) with π' gives

$$\log \frac{T_i^\star(\pi')(y \mid x)}{\pi'(y \mid x)} - \log \frac{T_i^\star(\pi')(y' \mid x)}{\pi'(y' \mid x)} = \Delta_i(x; y, y').$$

The same quantity is therefore obtained for π and π'. Since the two reference policies were arbitrary, the change induced by source i in the relative log-probability of y and y' is independent of the reference policy.

Finally, Assumption 3.1 also implies uniqueness of the optimizer in policy space. Let $\bar{\pi}$ be any exact optimizer of $\mathcal{L}_i(\cdot; \pi)$. Since $T_i^\star(\pi)$ attains the minimum value $\mathcal{R}_i(f_i^\star)$, the score $h_{\bar{\pi},\pi}$ must also minimize $\mathcal{R}_i$. By Assumption 3.1, there therefore exists $a : \mathcal{X} \to \mathbb{R}$ such that

$$h_{\bar{\pi},\pi}(x, y) = f_i^\star(x, y) + a(x).$$

Using the definition of the log-policy ratio,

$$\bar{\pi}(y \mid x) = \pi(y \mid x) \exp(f_i^\star(x, y) + a(x)) .$$

Summing over $y \in \mathcal{Y}$ and using that $\bar{\pi}(\cdot \mid x)$ is normalized gives

$$1 = \exp(a(x)) \sum_{y \in \mathcal{Y}} \pi(y \mid x) \exp(f_i^\star(x, y)) = \exp(a(x)) Z_{i,\pi}(x).$$

Hence

$$a(x) = -\log Z_{i,\pi}(x),$$

so that

$$\bar{\pi}(y \mid x) = \frac{\pi(y \mid x) \exp(f_i^\star(x, y))}{Z_{i,\pi}(x)} = T_i^\star(\pi)(y \mid x).$$

Thus the optimal score is identified only up to a prompt-dependent additive constant, while the corresponding optimal policy is unique. □

B.2 Proof of Proposition 3.1

Proposition B.1 (Restatement of Proposition 3.1)**.** *Let $S \subseteq \mathcal{N}$, let $\sigma = (\sigma_1, \ldots, \sigma_m)$ be any ordering of its $m = |S|$ sources, and consider the exact population sequential procedure*

$$\pi^{(0)} = \pi_0, \qquad \pi^{(k)} \in \arg\min_{\pi} \mathcal{L}_{\sigma_k}(\pi; \pi^{(k-1)}), \qquad k = 1, \ldots, m.$$

Under Assumption 3.1, its final policy is

$$\pi_S^\star(y \mid x) = \frac{\pi_0(y \mid x) \exp\big(\sum_{i \in S} f_i^\star(x, y)\big)}{\sum_{u \in \mathcal{Y}} \pi_0(u \mid x) \exp\big(\sum_{i \in S} f_i^\star(x, u)\big)}. \tag{14}$$

Consequently, the exact population policy depends only on the coalition S and not on the ordering σ.

Proof. For $k = 0, \dots, m$, let

$$S_k := \{\sigma_1, \dots, \sigma_k\}, \qquad S_0 := \emptyset,$$

and define

$$Z_{S_k}(x) := \sum_{u \in \mathcal{Y}} \pi_0(u \mid x) \exp\left(\sum_{i \in S_k} f_i^\star(x, u) \right). \tag{15}$$

With the convention that an empty sum is zero, $Z_{S_0}(x) = 1$.

We prove by induction on k that

$$\pi^{(k)}(y \mid x) = \frac{\pi_0(y \mid x) \exp\left(\sum_{i \in S_k} f_i^\star(x, y)\right)}{Z_{S_k}(x)}. \tag{16}$$

For $k = 0$, Equation (16) reduces to $\pi^{(0)} = \pi_0$, so the claim holds.

Suppose now that Equation (16) holds at stage $k - 1$. The induction hypothesis implies that $\pi^{(k-1)}$ has full support, so Lemma 3.1 applies with reference $\pi^{(k-1)}$. Moreover, as established in its proof, the population optimizer is unique in policy space. Therefore,

$$\pi^{(k)} = T_{\sigma_k}^\star(\pi^{(k-1)}).$$

It follows that

$$\begin{aligned}
\pi^{(k)}(y \mid x) &= \frac{\pi^{(k-1)}(y \mid x) \exp(f_{\sigma_k}^\star(x, y))}{\sum_{u \in \mathcal{Y}} \pi^{(k-1)}(u \mid x) \exp(f_{\sigma_k}^\star(x, u))} \\
&= \frac{\dfrac{\pi_0(y \mid x)}{Z_{S_{k-1}}(x)} \exp\left(\sum_{i \in S_k} f_i^\star(x, y)\right)}{\dfrac{1}{Z_{S_{k-1}}(x)} \sum_{u \in \mathcal{Y}} \pi_0(u \mid x) \exp\left(\sum_{i \in S_k} f_i^\star(x, u)\right)} \\
&= \frac{\pi_0(y \mid x) \exp\left(\sum_{i \in S_k} f_i^\star(x, y)\right)}{Z_{S_k}(x)}.
\end{aligned}$$

This establishes the induction claim.

Taking $k = m$ gives Equation (14). Since $S_m = S$, the final expression depends on the ordering σ only through the set of sources it contains. Because the sum $\sum_{i \in S} f_i^\star(x, y)$ is invariant to permutations of S, every ordering of the same coalition yields the same exact population policy. □

B.3 Proof of Proposition 3.2

Proposition B.2 (Restatement of Proposition 3.2)**.** *For each source i, let $\pi_i^\star = T_i^\star(\pi_0)$ denote the exact singleton policy, and for any coalition $S \subseteq \mathcal{N}$ define*

$$\widehat{s}_S^\star(x, y) = \log \pi_0(y \mid x) + \sum_{i \in S} \left[\log \pi_i^\star(y \mid x) - \log \pi_0(y \mid x) \right]. \tag{17}$$

Under Assumption 3.1, there exists a function $c_S : \mathcal{X} \to \mathbb{R}$, independent of the response y, such that

$$\widehat{s}_S^\star(x, y) = \log \pi_S^\star(y \mid x) + c_S(x). \tag{18}$$

Consequently, for any two responses $y, y' \in \mathcal{Y}$ to the same prompt x,

$$\log \pi_S^\star(y \mid x) - \log \pi_S^\star(y' \mid x) = \widehat{s}_S^\star(x, y) - \widehat{s}_S^\star(x, y'). \tag{19}$$

Proof. For each source i, define the singleton normalizer

$$Z_i(x) := \sum_{u \in \mathcal{Y}} \pi_0(u \mid x) \exp(f_i^\star(x, u)).$$

By Lemma 3.1,

$$\pi_i^\star(y \mid x) = \frac{\pi_0(y \mid x) \exp(f_i^\star(x, y))}{Z_i(x)}.$$

Taking logarithms gives

$$\log \pi_i^\star(y \mid x) - \log \pi_0(y \mid x) = f_i^\star(x, y) - \log Z_i(x). \tag{20}$$

Substituting Equation (20) into Equation (17) yields

$$\widehat{s}_S^\star(x, y) = \log \pi_0(y \mid x) + \sum_{i \in S} f_i^\star(x, y) - \sum_{i \in S} \log Z_i(x). \tag{21}$$

On the other hand, Proposition 3.1 gives

$$\log \pi_S^\star(y \mid x) = \log \pi_0(y \mid x) + \sum_{i \in S} f_i^\star(x, y) - \log Z_S(x), \tag{22}$$

where

$$Z_S(x) := \sum_{u \in \mathcal{Y}} \pi_0(u \mid x) \exp\left(\sum_{i \in S} f_i^\star(x, u) \right).$$

Comparing Equations (21) and (22) gives

$$\widehat{s}_S^\star(x, y) = \log \pi_S^\star(y \mid x) + \underbrace{\left[\log Z_S(x) - \sum_{i \in S} \log Z_i(x) \right]}_{=: c_S(x)}.$$

The term $c_S(x)$ depends on the coalition and the prompt, but not on the response. This proves Equation (18).

For $S = \emptyset$, $Z_S(x) = 1$ and the sum over $i \in S$ is empty, so $c_\emptyset(x) = 0$ and the same identity holds.

Finally, subtracting Equation (18) for two responses y and y' to the same prompt cancels $c_S(x)$, yielding Equation (19). □

For completeness, the fixed-candidate normalization identity used immediately after Proposition 3.2 follows from the same result. For any nonempty candidate set $\mathcal{C}(x) \subseteq \mathcal{Y}$, exponentiating Equation (18) gives

$$\exp(\widehat{s}_S^\star(x, y)) = \exp(c_S(x)) \pi_S^\star(y \mid x).$$

The factor $\exp(c_S(x))$ is common to every response associated with prompt x and therefore cancels under normalization:

$$\frac{\exp(\widehat{s}_S^\star(x, y))}{\sum_{u \in \mathcal{C}(x)} \exp(\widehat{s}_S^\star(x, u))} = \frac{\pi_S^\star(y \mid x)}{\sum_{u \in \mathcal{C}(x)} \pi_S^\star(u \mid x)}.$$

Thus the reconstruction preserves normalized distributions over a fixed candidate set, as well as all pairwise score differences, rankings, and argmax decisions within that set.

Likewise, if the evaluation protocol $\mathcal{V}$ satisfies the shift invariance condition (6), then

$$\mathcal{V}(\widehat{s}_S^\star) = \mathcal{V}(\log \pi_S^\star + c_S) = \mathcal{V}(\log \pi_S^\star) = v(S).$$

This is the identity used in Section 3.4 to recover every exact coalition utility from the base policy and the n singleton policies, without any additional coalition-specific alignment.

C EXPERIMENTAL DETAILS

This appendix provides the complete experimental protocol used in Sections 4 and 5 and in Appendix D.

C.1 PREFERENCE DATA

UltraFeedback sources. We use UltraFeedback [9] and define one preference source from each of the following nine prompt provenances: `evol_instruct`, `false_qa`, `flan_v2_cot`, `flan_v2_flan2021`, `flan_v2_niv2`, `flan_v2_p3`, `sharegpt`, `truthful_qa`, and `ultrachat`. These nine sources constitute the player set $\mathcal{N}$ throughout the experiments.

For each UltraFeedback prompt, we retain the highest-scoring valid completion as the preferred response y^+ and the lowest-scoring valid completion as the rejected response y^-. Examples with fewer than two valid scored completions are discarded. Each resulting example is therefore a preference triple (x, y^+, y^-) associated with one source.

Training and final evaluation split. We first construct the complete normalized dataset containing the nine sources and shuffle it once with a fixed seed. We then allocate 90% of the examples to the training pool and hold out the remaining 10% for final evaluation. The two sets are disjoint at the prompt-example level, and the final evaluation pool is never used for training or checkpoint selection.

The 90% training pool is partitioned by provenance to obtain one dataset per source. For each source i, this source-specific pool is further split into 75% examples used for optimization and 25% examples used for checkpoint selection. We do not cap or rebalance the source training sets: their original relative sizes are preserved.

Construction of the common evaluation dataset. The common evaluation dataset is constructed exclusively from the held-out 10% split. We first partition this split by source, retain at most 256 preference triples from each source, concatenate the resulting source-specific subsets, and shuffle the concatenated dataset with a fixed seed. We refer to this dataset as `uf9_eval`. No example in `uf9_eval` is used for training or optimization.

For the finite-training comparisons in Section 4, we use the same fixed set $\mathcal{E} = \{(x_j, y_j^+, y_j^-)\}_{j=1}^{M}$, $M = 512$, drawn from `uf9_eval`. The same 512 preference triples are used for all models, coalitions, source orders, reconstructions, and repeated training runs.

C.2 MODELS AND OPTIMIZATION

Base policies. We evaluate four instruction-tuned base policies: `HuggingFaceTB/SmolLM-135M-Instruct`, `Qwen/Qwen2.5-0.5B-Instruct`, `meta-llama/Llama-3.2-1B-Instruct`, and `microsoft/Phi-3.5-mini-instruct`. We refer to them as SmolLM, Qwen2.5, Llama3.2, and Phi-3.5, respectively.

IPO training. The main finite-training experiments use IPO [3] with $\beta = 0.1$ for all four base policies. Each source is optimized for at most 3 epochs. We use a learning rate of 2×10^{-5}, per-device batch size 4, gradient accumulation over 4 steps, bfloat16 precision, and a maximum sequence length of 1024. Except when the optimization seed is deliberately varied to estimate retraining variability, all compared runs use the same seed and identical optimization settings.

DPO training. The complementary DPO experiments use the sigmoid DPO loss and the same optimization settings as above: a learning rate of 2×10^{-5}, per-device batch size 4, gradient accumulation over 4 steps, a maximum sequence length of 1024, bfloat16 precision, and an epoch ceiling of 3. The DPO temperature is $\beta = 0.1$ for SmolLM and Qwen2.5. All DPO stages use the same stage-wise checkpoint-selection procedure described below.

C.3 CHECKPOINT SELECTION

Checkpoint selection is performed separately for every source optimization. This applies both to singleton training and to *every stage* of a sequential alignment run.

Suppose a stage optimizes source i from reference policy ρ. Candidate checkpoints are evaluated only on the 25% validation subset of source i. For checkpoint selection, completion log-probabilities are length-normalized: for a response $y = (y_1, \ldots, y_T)$,

$$\bar{\ell}_\pi(y \mid x) = \frac{1}{T} \sum_{t=1}^{T} \log \pi(y_t \mid x, y_{<t}).$$

We then define the update margin relative to the current reference as

$$m_{\pi,\rho}(x, y^+, y^-) = \left[\bar{\ell}_\pi(y^+ \mid x) - \bar{\ell}_\rho(y^+ \mid x)\right] - \left[\bar{\ell}_\pi(y^- \mid x) - \bar{\ell}_\rho(y^- \mid x)\right].$$

The primary checkpoint-selection criterion is the fraction of validation examples for which $m_{\pi,\rho} > 0$, i.e., the fraction for which the update moves the preferred response upward relative to the rejected response. Ties are broken first by the improvement in preference accuracy relative to the reference policy and then by the mean update margin.

In a sequential run, once a checkpoint has been selected at stage k, this checkpoint becomes both the initialization and the reference policy for stage $k+1$. Hence checkpoint selection is performed anew after every source in the sequence, using the held-out validation examples of the source currently being optimized.

Length normalization is used only for checkpoint selection. All sequence-level quantities used in the final finite-training evaluation and in the coalition reconstruction use ordinary, non-length-normalized complete-response log-probabilities.

C.4 SINGLETON AND SEQUENTIAL TRAINING

Singleton models. For every source $i \in \mathcal{N}$, we train a singleton policy π_i using only the training portion of source i. Both the initialization and the reference policy are the corresponding base policy π_0. The retained singleton is the checkpoint selected according to the procedure above.

Sequential coalition models. For a coalition S and an ordering $\sigma = (\sigma_1, \ldots, \sigma_{|S|})$, we start from $\pi^{(0)} = \pi_0$. At stage k, source σ_k is optimized with $\pi^{(k-1)}$ used both as initialization and reference, i.e., the selected checkpoint from the previous stage is used both to initialize the trainable policy and as the fixed reference policy in the DAA objective:

$$\pi^{(k)} \leftarrow \mathrm{DAA}\Big(D^{\mathrm{train}}_{\sigma_k}; \mathrm{init} = \pi^{(k-1)}, \mathrm{ref} = \pi^{(k-1)}\Big).$$

Here, $\pi^{(k)}$ denotes the checkpoint selected on the validation subset of source σ_k, rather than necessarily the final checkpoint of the three-epoch training budget. The next stage starts from this selected checkpoint.

C.5 FINITE-TRAINING VALIDATION

Evaluated coalitions and source orders. For each base model, we evaluate coalitions of sizes $|S| \in \{3, 6, 9\}$. For every coalition, three prespecified source orderings are trained under otherwise identical conditions. All three pairwise comparisons between these orderings are included in the order-robustness evaluation.

Coalitions, source orders, and repetitions. For the finite-training validation, we consider three nested coalitions $S_3, S_6, S_9 \subseteq \mathcal{N}$ of respective sizes 3, 6, and 9:

$$S_3 = \{\texttt{evol_instruct}, \texttt{sharegpt}, \texttt{ultrachat}\},$$

$$S_6 = \{\texttt{evol_instruct}, \texttt{false_qa}, \texttt{flan_v2_flan2021}, \texttt{sharegpt}, \texttt{truthful_qa}, \texttt{ultrachat}\},$$

$$S_9 = \mathcal{N}.$$

We fix three global permutations of the complete set of nine sources:

$$\sigma_1 = \big(\texttt{evol_instruct}, \texttt{false_qa}, \texttt{flan_v2_cot}, \texttt{flan_v2_flan2021}, \texttt{flan_v2_niv2}, \texttt{flan_v2_p3}, \texttt{sharegpt}, \texttt{truthful_qa}, \texttt{ultrachat}\big),$$

$$\sigma_2 = \big(\texttt{ultrachat}, \texttt{truthful_qa}, \texttt{sharegpt}, \texttt{flan_v2_p3}, \texttt{flan_v2_niv2}, \texttt{flan_v2_flan2021}, \texttt{flan_v2_cot}, \texttt{false_qa}, \texttt{evol_instruct}\big),$$

$$\sigma_3 = \big(\texttt{sharegpt}, \texttt{flan_v2_flan2021}, \texttt{evol_instruct}, \texttt{ultrachat}, \texttt{flan_v2_cot}, \texttt{truthful_qa}, \texttt{false_qa}, \texttt{flan_v2_p3}, \texttt{flan_v2_niv2}\big).$$

For a coalition S, we denote by $\sigma_k|_S$ the restriction of σ_k to the sources contained in S, obtained by deleting all sources outside S while preserving the relative order of the remaining ones. Accordingly, the three orders used for the size-3 coalition are

$$\sigma_1|_{S_3} = (\texttt{evol_instruct}, \texttt{sharegpt}, \texttt{ultrachat}),$$
$$\sigma_2|_{S_3} = (\texttt{ultrachat}, \texttt{sharegpt}, \texttt{evol_instruct}),$$
$$\sigma_3|_{S_3} = (\texttt{sharegpt}, \texttt{evol_instruct}, \texttt{ultrachat}),$$

while for the size-6 coalition they are

$$\sigma_1|_{S_6} = (\texttt{evol_instruct}, \texttt{false_qa}, \texttt{flan_v2_flan2021}, \texttt{sharegpt}, \texttt{truthful_qa}, \texttt{ultrachat}),$$
$$\sigma_2|_{S_6} = (\texttt{ultrachat}, \texttt{truthful_qa}, \texttt{sharegpt}, \texttt{flan_v2_flan2021}, \texttt{false_qa}, \texttt{evol_instruct}),$$
$$\sigma_3|_{S_6} = (\texttt{sharegpt}, \texttt{flan_v2_flan2021}, \texttt{evol_instruct}, \texttt{ultrachat}, \texttt{truthful_qa}, \texttt{false_qa}).$$

For $S_9 = \mathcal{N}$, the three orders are simply $\sigma_1, \sigma_2, \sigma_3$.

For each base model and each coalition $S \in \{S_3, S_6, S_9\}$, we train one sequential coalition model under each of the three orders $\sigma_1|_S$, $\sigma_2|_S$, and $\sigma_3|_S$. Order sensitivity is evaluated using all three pairwise comparisons,

$$(\sigma_1, \sigma_2), \qquad (\sigma_1, \sigma_3), \qquad (\sigma_2, \sigma_3),$$

yielding three values of d_{order} for each model–coalition pair. Reconstruction fidelity is evaluated separately against the explicitly trained coalition produced under each of the three orders, yielding three values of d_{recon} for the same model–coalition pair.

We designate $\sigma_{\text{ref}} = \sigma_1$ as the global reference ordering. For each model and coalition S, we repeat the complete sequential training procedure following $\sigma_{\text{ref}}|_S$ with $R = 3$ independent optimization seeds. The coalition, source order, data splits, hyperparameters, and checkpoint-selection procedure are held fixed across these repetitions. The resulting three runs are used to estimate the same-procedure retraining floor $d_{\text{floor}}(S)$, which is shared by all order and reconstruction comparisons associated with that model–coalition pair.

Complete-response preference gaps. All finite-training discrepancies are evaluated on the common $M = 512$ examples in $\mathcal{E}$. For a policy π, we define

$$G_\pi^{(j)} = \log \pi(y_j^+ \mid x_j) - \log \pi(y_j^- \mid x_j). \tag{23}$$

Here, $\log \pi(y \mid x)$ is the complete-response log-probability, obtained by summing token log-probabilities over the response tokens. No length normalization is applied.

For two vectors of preference gaps G and G', we use

$$d(G, G') = \frac{1}{M} \sum_{j=1}^{M} \left| G^{(j)} - G'^{(j)} \right|. \tag{24}$$

Order discrepancy. For two source orderings σ and τ of the same coalition,

$$d_{\text{order}}(S; \sigma, \tau) = d\big(G_{\pi_{S,\sigma}}, G_{\pi_{S,\tau}}\big). \tag{25}$$

Reconstruction discrepancy. Given the trained singleton policies, the finite-training reconstruction is

$$\widehat{s}_S(x, y) = \log \pi_0(y \mid x) + \sum_{i \in S} [\log \pi_i(y \mid x) - \log \pi_0(y \mid x)]. \tag{26}$$

The corresponding reconstructed preference gap is

$$\widehat{G}_S^{(j)} = \widehat{s}_S(x_j, y_j^+) - \widehat{s}_S(x_j, y_j^-), \tag{27}$$

and, for a sequential coalition trained under ordering σ,

$$d_{\text{recon}}(S; \sigma) = d\Big(\widehat{G}_S, G_{\pi_{S,\sigma}}\Big). \tag{28}$$

Retraining floor. For every model and coalition S, we fix one reference ordering σ_{ref} and train this same ordered coalition independently with $R = 3$ optimization seeds, while keeping the data, order, and all other experimental choices fixed. The retraining floor is

$$d_{\text{floor}}(S) = \frac{2}{R(R-1)} \sum_{1 \le r < r' \le R} d\Big(G_{\pi^{(r)}_{S,\sigma_{\text{ref}}}}, G_{\pi^{(r')}_{S,\sigma_{\text{ref}}}}\Big). \tag{29}$$

Thus, with $R = 3$, $d_{\text{floor}}(S)$ is the mean of the three pairwise distances between independent repetitions of the same ordered training procedure. The same value $d_{\text{floor}}(S)$ is used to contextualize every order and reconstruction discrepancy associated with the corresponding model and coalition.

C.6 Shapley valuation

Candidate pools. We use the prompts $\mathcal{X}_{\mathcal{E}} = \{x_j\}_{j=1}^M$ from the common held-out evaluation set. For each base policy π_0 and each prompt x, we sample once a fixed pool of $K = 16$ complete responses,

$$C(x) = \{y_1, \ldots, y_{16}\}, \qquad y_k \sim \pi_0(\cdot \mid x).$$

Generation uses fixed random seeds and a maximum of 64 newly generated tokens per response. Consequently, for a given base model and prompt, the candidate pool is independent of the coalition being evaluated and is reused for all 2^9 coalitions and all reward models.

We never sample from a reconstructed next-token coalition policy. For each coalition S, we instead score every fixed candidate using

$$\widehat{s}_S(x, y) = \log \pi_0(y \mid x) + \sum_{i \in S} [\log \pi_i(y \mid x) - \log \pi_0(y \mid x)], \tag{30}$$

and select

$$\widehat{y}_S(x) = \arg\max_{y \in C(x)} \widehat{s}_S(x, y). \tag{31}$$

The same candidate responses are therefore compared for every coalition.

Reward models. We evaluate each selected response using three external reward models: `Skywork/Skywork-Reward-V2-Qwen3-0.6B`, `Ray2333/GRM-Llama3.2-3B-rewardmodel-ft`, and `openbmb/UltraRM-13b`. We denote these models by r_1, r_2, and r_3, respectively.

Because the numerical scales of the reward models differ and to make comparisons across reward models visually interpretable, each reward is standardized over the same set of evaluated prompt-candidate pairs:

$$\widetilde{r}(x, y) = \frac{r(x, y) - \mu_r}{\sigma_r}, \tag{32}$$

where μ_r and σ_r are the mean and standard deviation of $r(x, y)$ over all $x \in \mathcal{X}_{\mathcal{E}}$ and all $y \in C(x)$. These statistics are computed once for each reward model and kept fixed for all coalitions.

The utility of coalition S is

$$\widehat{v}_r(S) = \frac{1}{|\mathcal{X}_{\mathcal{E}}|} \sum_{x \in \mathcal{X}_{\mathcal{E}}} \widetilde{r}(x, \widehat{y}_S(x)). \tag{33}$$

Shapley computation. With $n = 9$, we evaluate all

$$2^9 = 512$$

coalitions. For each reward model r, the Shapley value of source i is therefore computed exactly for the reconstructed game:

$$\phi_i(\widehat{v}_r) = \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \left[\widehat{v}_r(S \cup \{i\}) - \widehat{v}_r(S)\right]. \tag{34}$$

No Monte Carlo approximation is used in the Shapley summation. The approximation in this paper instead replaces an explicitly trained coalition with its finite-training reconstruction from singleton policies.

Because utilities are computed from standardized rewards, $\phi_i(\widehat{v}_r)$ is expressed in standard-deviation units of reward model r.

D ADDITIONAL EXPERIMENTS

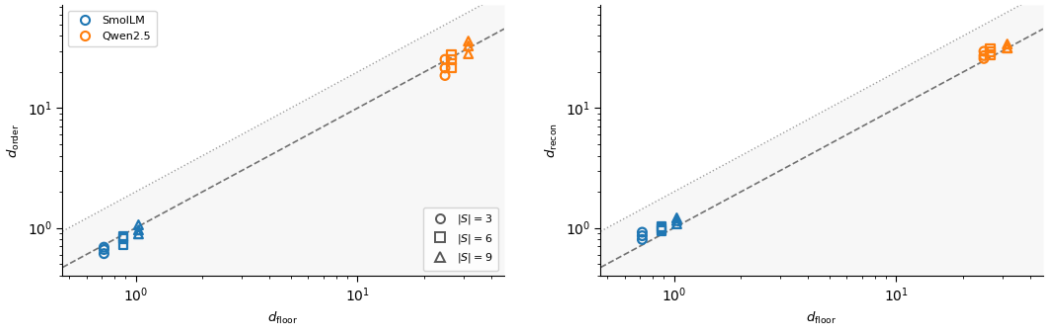
This appendix reports two complementary experiments. First, we repeat the finite-training validation in Section 4 using DPO instead of IPO to verify that the empirical behavior supporting our reconstruction is not specific to the alignment loss used. Second, we compare sequential alignment with conventional pooled training on the same collections of sources. The latter comparison is not implied by our theory since sequential and pooled optimization correspond to different finite-training procedures. Rather, it serves as a sanity check that processing preference sources sequentially does not lead to substantially different alignment behavior from training jointly on their union.

D.1 COMPLEMENTARY EXPERIMENTS WITH DPO

The theoretical results of Section 3 apply to the class of Direct Alignment Algorithms described in Section 2, rather than specifically to IPO. Our main finite-training experiments instantiate this class with IPO. We therefore repeat the same validation with Direct Preference Optimization (DPO) [32] for two of the base models considered in the main experiments, `SmolLM-135M-Instruct` and `Qwen2.5-0.5B-Instruct`.

We keep the experimental protocol of Section 4 unchanged. All details are given in Appendix C. The only substantive change is the direct-alignment loss, since DPO uses $g(t) = -\log \sigma(t)$ in place of the IPO objective.

Figure 3 reports the order and reconstruction discrepancies against the corresponding retraining floors. The plotting convention matches Figure 1.



(a) Order robustness

(b) Reconstruction fidelity

Figure 3: Finite-training validation with DPO for `SmolLM-135M-Instruct` and `Qwen2.5-0.5B-Instruct`. Left: source-order discrepancy d_{order}. Right: reconstruction discrepancy d_{recon}, both plotted against the corresponding retraining floor d_{floor}. The dashed diagonal corresponds to equality with the retraining floor, while the dotted line marks twice that floor.

The DPO results exhibit the same qualitative pattern as the IPO experiments in Section 4. Across both base models and the tested coalition sizes, source-order and reconstruction discrepancies remain of the same order as the corresponding retraining variability. Together with the IPO results, these experiments provide empirical evidence that the finite-training behavior underlying our reconstruction remains consistent with the broader class of DAAs covered by our theoretical analysis under Assumption 3.1.

D.2 SEQUENTIAL VERSUS POOLED PREFERENCE OPTIMIZATION

We refer to *pooled training* as the conventional procedure in which the preference examples from all sources in a coalition S are combined into a single training set and optimized jointly in one run, with the base policy π_0 used as the reference throughout training.

Our theory concerns sequential optimization and does not imply that sequential and pooled training are identical under finite optimization. Even under exact optimization, sequentially optimizing the

Coalition	$d_{\mathrm{seq/pool}}$	d_{within}	$\rho_{\mathrm{seq/pool}}$	Ranking disagreement
S_3	0.73	0.60	1.20	1.3%
S_6	1.01	0.78	1.28	2.2%
S_9	1.30	0.90	1.43	3.0%

Table 1: Sequential versus pooled IPO training on `SmolLM-135M-Instruct`. $d_{\mathrm{seq/pool}}$ is the largest discrepancy over the three tested sequential source orders, while d_{within} measures the largest same-procedure retraining variability. Ranking disagreement is the fraction of held-out preference pairs on which the two procedures prefer different responses.

source-specific objectives need not recover the minimizer of their pooled objective. The experiment instead asks whether, in the finite-training regime, replacing pooled training with the sequential procedure underlying our reconstruction substantially changes the resulting preference behavior.

We use `HuggingFaceTB/SmolLM-135M-Instruct` with IPO and consider the same coalitions S_3, S_6, and S_9 as in the finite-training validation. For each coalition, pooled training performs a single IPO optimization on the union of its source datasets, using π_0 as the fixed reference. Sequential training processes the same datasets source by source, using each updated policy as the reference for the next stage. We evaluate the three sequential source orders defined in Appendix C. Both procedures use the same data and training hyperparameters, and each configuration is repeated with $R = 3$ independent optimization seeds.

We compare the resulting policies directly on the common held-out preference set, without using an external reward model. For a policy π, let

$$G_\pi(x, y^+, y^-) = \log \pi(y^+ \mid x) - \log \pi(y^- \mid x),$$

and recall the discrepancy

$$d(\pi, \pi') = \frac{1}{M} \sum_{j=1}^{M} \left| G_\pi^{(j)} - G_{\pi'}^{(j)} \right|.$$

For each coalition, we report the largest sequential-pooled discrepancy over the three sequential orders. To make this quantity interpretable, we compare it with the variability obtained by independently repeating the same training procedure. Specifically, we define d_{within} as the largest mean same-procedure discrepancy across the pooled and sequential configurations, and report

$$\rho_{\mathrm{seq/pool}} = \frac{d_{\mathrm{seq/pool}}}{d_{\mathrm{within}}}.$$

Thus, $\rho_{\mathrm{seq/pool}} = 1$ means that the difference between sequential and pooled training equals ordinary retraining variability. We also report the fraction of held-out preference pairs on which the two procedures induce opposite rankings.

Table 1 shows that the discrepancy between sequential and pooled training remains close to the intrinsic variability of retraining across all three coalition sizes. The ratio $\rho_{\mathrm{seq/pool}}$ ranges from 1.20 to 1.43. The discrepancy increases moderately with coalition size, but this increase is not accompanied by a substantial change in the induced preference decisions. In fact, sequential and pooled policies disagree on only 1.3% to 3.0% of held-out preference pairs. Thus, for `SmolLM-135M-Instruct` in the regime considered here, sequential training produces preference behavior that is close to pooled IPO relative to the stochastic variability of training itself, while remaining measurably distinct from it.

E ADDITIONAL FIGURES

This appendix complements Figure 2 by reporting the same Shapley-value representation for the other base models considered in our experiments. Figures 4-6 use exactly the evaluation protocol of Section 5. Only the base model and corresponding singleton policies change.

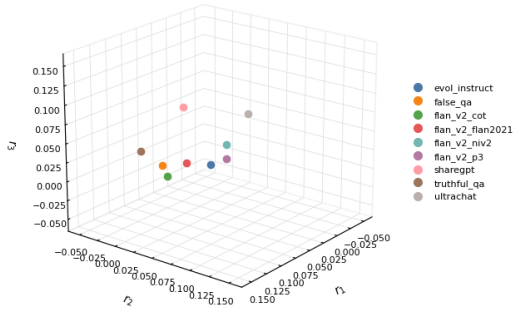


Figure 4: Shapley values of the $n = 9$ preference sources for SmolLM under three standardized reward models. Each source i is represented by $(\phi_i(\widehat{v}_{r_1}), \phi_i(\widehat{v}_{r_2}), \phi_i(\widehat{v}_{r_3})$. Along axis r_k, coordinates are expressed in standard-deviation units of the corresponding reward score.

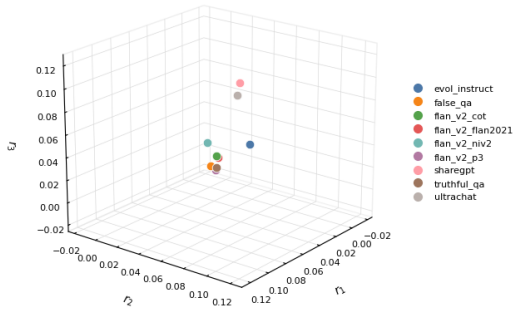


Figure 5: Shapley values of the $n = 9$ preference sources for Llama3.2 under three standardized reward models. Each source i is represented by $(\phi_i(\widehat{v}_{r_1}), \phi_i(\widehat{v}_{r_2}), \phi_i(\widehat{v}_{r_3})$. Along axis r_k, coordinates are expressed in standard-deviation units of the corresponding reward score.

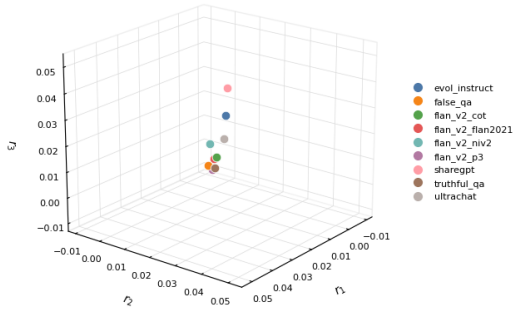


Figure 6: Shapley values of the $n = 9$ preference sources for Phi-3.5 under three standardized reward models. Each source i is represented by $(\phi_i(\widehat{v}_{r_1}), \phi_i(\widehat{v}_{r_2}), \phi_i(\widehat{v}_{r_3})$. Along axis r_k, coordinates are expressed in standard-deviation units of the corresponding reward score.

F ADDITIONAL DISCUSSIONS

F.1 EXTENSION TO SEMIVALUE-BASED DATA VALUATION

Our main analysis focuses on the Shapley value, but the reduction in alignment cost is not specific to the Shapley weighting. It applies directly to the broader family of semivalues introduced in Section A.3.

Let $\omega = (\omega_0, \ldots, \omega_{n-1})$ be a probability distribution over coalition sizes, with $\omega_k \geq 0$ and $\sum_{k=0}^{n-1} \omega_k = 1$. The corresponding semivalue of source i can be written as

$$\psi_i^{\omega}(v) = \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{\omega_{|S|}}{\binom{n-1}{|S|}} \left[v(S \cup \{i\}) - v(S) \right]. \tag{35}$$

Thus, ω_k determines the total weight assigned to marginal contributions where source i is added to a coalition of size k, with this weight shared uniformly among coalitions of that size. The Shapley value is recovered by $\omega_k = 1/n$ for every k. Other choices recover alternative semivalues. For example, the Banzhaf value [4, 40] corresponds to

$$\omega_k = \frac{\binom{n-1}{k}}{2^{n-1}},$$

which assigns equal weight to every coalition $S \subseteq \mathcal{N} \setminus \{i\}$.

The reconstruction developed in Section 3 is independent of these weights. Under the exact population assumptions of that section, for every coalition S the singleton reconstruction recovers the complete-response log-probabilities needed to evaluate $v(S)$, up to a prompt-dependent additive constant. Whenever the evaluation procedure is invariant to this constant, the reconstructed utility satisfies

$$\widehat{v}(S) = v(S) \qquad \text{for every } S \subseteq \mathcal{N}.$$

Substituting these reconstructed utilities into Equation (35) therefore gives

$$\psi_i^{\omega}(\widehat{v}) = \psi_i^{\omega}(v)$$

for every source i and every valid choice of semivalue weights ω.

Consequently, changing from the Shapley value to another semivalue requires no additional alignment runs. The same base policy and the same n singleton policies are sufficient. Only the weights used to aggregate coalition marginal contributions change. As for exact Shapley computation, evaluating an arbitrary semivalue exactly can still require utilities for exponentially many coalitions, but these utilities can be obtained through the same reconstruction rather than through coalition-specific alignment. Sampling-based estimators of semivalues can likewise be combined with our reconstruction when evaluating all coalitions is undesirable.

One distinction is important for interpretation. The efficiency identity used for the Shapley value,

$$\sum_{i \in \mathcal{N}} \phi_i(v) = v(\mathcal{N}) - v(\emptyset),$$

does not hold for a general semivalue. Alternative semivalue weights therefore change not only the relative emphasis placed on different coalition sizes but also the axiomatic interpretation of the resulting data values. Our experiments focus on Shapley values because of this efficiency property. The choice among alternative semivalues is orthogonal to the coalition reconstruction developed in this work.

F.2 FROM COMPLETE-RESPONSE SCORES TO NEXT-TOKEN PROBABILITIES

Our reconstruction is defined at the level of complete responses. This section makes precise why this does not directly provide the next-token probabilities required for autoregressive generation.

Consider first the exact-optimization setting. By Proposition 3.2, for every coalition S,

$$\widehat{s}_S^{\star}(x, y) = \log \pi_S^{\star}(y \mid x) + c_S(x),$$

where $c_S(x)$ does not depend on the response y. Hence $\widehat{s}^\star_S$ determines the complete-response distribution $\pi^\star_S(\cdot \mid x)$ up to its normalizing constant.

Let u denote a response prefix, and let

$$\mathcal{Y}(u) = \{y \in \mathcal{Y} : \ y \text{ has prefix } u\}.$$

For a token a that can follow u, the next-token probability induced by the complete-response distribution is

$$\pi^\star_S(a \mid x, u) = \frac{\sum_{y \in \mathcal{Y}(ua)} \exp(\widehat{s}^\star_S(x, y))}{\sum_b \sum_{y \in \mathcal{Y}(ub)} \exp(\widehat{s}^\star_S(x, y))}. \tag{36}$$

The prompt-dependent constant $c_S(x)$ and the global sequence-level normalizer cancel from this ratio. Nevertheless, Equation (36) requires summing over all complete continuations of each possible next token. Thus, knowing how to score an individual complete response does not by itself provide an efficient way to compute the corresponding next-token distribution.

The distinction can also be seen directly from the autoregressive factorization of the reconstructed score. From Equation (7),

$$\exp(\widehat{s}_S(x, y)) = \pi_0(y \mid x) \prod_{i \in S} \frac{\pi_i(y \mid x)}{\pi_0(y \mid x)}.$$

Writing $y = (y_1, \ldots, y_T)$ and factorizing each model autoregressively gives

$$\exp(\widehat{s}_S(x, y)) = \prod_{t=1}^{T} w_S(y_t \mid x, y_{<t}),$$

where

$$w_S(a \mid x, u) = \pi_0(a \mid x, u) \prod_{i \in S} \frac{\pi_i(a \mid x, u)}{\pi_0(a \mid x, u)}. \tag{37}$$

It may therefore be tempting to obtain a next-token policy by normalizing $w_S(\cdot \mid x, u)$ locally. In general, however, this does not recover the conditional distribution associated with the complete-response scores.

To see why, define the continuation partition function

$$Z_S(x, u) = \sum_{y \in \mathcal{Y}(u)} \prod_{t > |u|} w_S(y_t \mid x, y_{<t}),$$

with the natural convention that $Z_S(x, u) = 1$ when u is already a complete response. The next-token conditional corresponding to the globally normalized complete-response scores is then

$$q_S(a \mid x, u) = \frac{w_S(a \mid x, u)\, Z_S(x, ua)}{\sum_b w_S(b \mid x, u)\, Z_S(x, ub)}. \tag{38}$$

By contrast, locally combining and normalizing the singleton next-token probabilities would give

$$\widetilde{q}_S(a \mid x, u) = \frac{w_S(a \mid x, u)}{\sum_b w_S(b \mid x, u)}. \tag{39}$$

Equations (38) and (39) coincide only in special cases, for example, when the continuation partition $Z_S(x, ua)$ is the same for every possible next token a. There is no reason for this condition to hold for a general autoregressive language model since different next tokens lead to different prefixes and therefore to different distributions over all subsequent continuations.

This explains the scope of our reconstruction result. Evaluating $\widehat{s}_S(x, y)$ for a fixed complete response y only requires teacher-forced evaluation of the base and singleton models along that response. Likewise, comparing or selecting among a fixed finite set of candidate responses requires

no sequence-level normalization. In contrast, sampling autoregressively from the complete-response distribution represented by $\widehat{s}_S$ would require computing, or approximating, the continuation partition functions in Equation (38) at every decoding step. For a general language model, these quantities involve summation over the tree of possible future continuations and cannot be obtained by simply normalizing the singleton probabilities at the current token.

For this reason, the present work uses the reconstruction only to score fixed complete responses and does not claim to reconstruct a generative coalition policy. Developing tractable approximations of the continuation marginalization required for direct generation is left for future work.