RETHINKING LEAST-CORE COMPUTATION IN CONTEXTUAL-DISTRACTOR GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Game-theoretic attribution explains a model by assigning credit to its features or training examples. The least core has attracted interest as an alternative to Shapley-style averaging because it can expose players that cause substantial harm in rare, high-value contexts. However, least-core allocations are generally nonunique, and the choice of allocation can affect the resulting explanation. In this study, we investigate how payoff selection and coalition sampling affect least-core attribution. Our experiments show that selector choice matters for distinguishing useful and harmful contributions, and that sampling can degrade harmful-player identification across the tested selectors even when useful players remain well identified. These observations motivate efficient computation with all coalition constraints and a well-defined selector. We introduce entropic least core (ELC), a smooth approximation whose unique minimizer follows a continuous path along the temperature to the nucleolus, a classical refinement of the least core. Our experiments show that ELC approximates the nucleolus faster than an LP-based nucleolus solver while retaining small payoff errors, with further GPU acceleration at larger problem sizes. In the tested full-coalition contextual-distractor games, ELC matches the minimum-norm selector in identification accuracy and more accurately ranks distractors by harm.

1 INTRODUCTION

A fundamental building block of machine-learning model explanation is scoring the contribution of each feature, training example, or agent to the model's output. Game-theoretic attribution has provided a framework for measuring contributions through a coalitional game: *players* work together as a coalition, and the central task is to allocate the resulting value fairly among the players. In particular, the Shapley value (Shapley, 1953) is a widely used index. The Shapley value credits each player by its average marginal contribution: it measures the change in coalition value when the player is added, and takes a weighted average over all coalitions not containing that player. Applications include feature attribution, data valuation, and computer-vision explanations (Štrumbelj & Kononenko, 2010; Lundberg & Lee, 2017; Sundararajan et al., 2017; Ghorbani & Zou, 2019; Covert et al., 2020; Jethani et al., 2022; Sumiyasu et al., 2024; Yamauchi et al., 2026; Blücher et al., 2022).

The least core (Maschler et al., 1979) has recently been studied as a useful game-theoretic index for model explanation (Yan & Procaccia, 2021; Gemp et al., 2024; Nagarajan et al., 2026; Harris et al., 2026). Rather than averaging the marginal contribution, the least core seeks a stable payoff allocation that minimizes the maximum underpayment to coalitions. The critical difference from the Shapley value has recently been formalized by Nagarajan et al. (2026) through *contextual distractors*, players that have a minor positive impact in most coalitions but cause substantial harm in rare, high-value coalitions. In their experiments, the least core assigned negative payoffs to them by accounting for the worst case, whereas the Shapley value failed, assigning positive payoffs because averaging lets frequent minor positive contributions outweigh rare but substantial harm. This distinction is relevant to understanding model failures in rare, high-impact contexts.

However, the aforementioned studies provide limited discussion of the choice of *selector* for the least core. Least-core computation is formulated as a linear program (LP) with exponentially many constraints, each corresponding to a coalition. The minimizing payoff vectors are generally not unique, forming a polytope. A selector is a map that chooses one payoff allocation from the polytope, and

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.

this choice affects the resulting explanation. However, many works use the minimum-norm selector as a de facto standard (Yan & Procaccia, 2021; Harris et al., 2026; Nagarajan et al., 2026), while its suitability for contextual-distractor identification relative to other selectors remains insufficiently examined. Approximate computation methods (Yan & Procaccia, 2021; Gemp et al., 2024) do not discuss the final payoff allocation, either; Yan & Procaccia (2021) only bounds the optimality gap of minimum value, not the minimizers. Gemp et al. (2024) did not specify the selector, and thus the final payoff allocation is implicitly determined by the optimization process and its parameters.

In this study, we investigate least-core computation with a focus on payoff selection. Particularly, we focus on the effect of the selector choice and coalition sampling on the contextual-distractor setting. Our key observations are twofold: (i) the choice of selector is essential for successful retrieval of contextual distractors, and (ii) coalition sampling particularly degrades their retrieval for standard selectors. For the former, we found the nucleolus selector to be the most successful, but expensive to compute. The latter is particularly important because existing acceleration methods (Yan & Procaccia, 2021; Gemp et al., 2024) rely on coalition sampling. These observations motivate a more tractable approximation of the nucleolus with full constraints. We introduce the *entropic least core* (ELC), a smoothed approximation of the least core that replaces the maximum excess with a log-sum-exp objective. At temperature $\eta > 0$, the ELC objective is convex and differentiable, with a unique minimizer, and can be optimized by projected gradient descent with GPU acceleration.

Our contributions are as follows:

- We revisit least-core computation from the payoff allocation perspective and provide empirical observations to inform its design. We show the substantial impact of selector choice and the potential unreliability of coalition sampling for retrieving contextual distractors.
- We propose the η-entropic least core (η-ELC) as a smoothed approximation of the least core. Its unique payoff varies continuously with temperature and converges to the nucleolus as $\eta \to 0$. Our experiments show that ELC approximates the nucleolus with small payoff errors in less solver time than the successive-LP nucleolus solver.
- Our experiments compare Shapley attribution and least-core methods with different selectors. (i) In the tested games, all methods identify useful players perfectly with full coalitions. Under sampling, useful-player identification remains comparatively strong, although Shapley performance declines. (ii) With full coalitions, the minimum-norm selector and ELC both attain perfect average precision for contextual-distractor identification, but ELC more accurately ranks distractors by harm. ELC thus provides more fine-grained attribution in these games.

2 Related work

The Shapley value (Shapley, 1953) is a widely used game-theoretic measure of contribution in model explanation (Štrumbelj & Kononenko, 2010; Lundberg & Lee, 2017),[1] whereas the least core (Maschler et al., 1979) has received comparatively limited attention. Yan & Procaccia (2021) introduced the core and least core for model explainability. Toward computational tractability, they showed that solving an LP with sampled coalition constraints may attain *probably approximate* least core, i.e., good approximation of the minimal maximal excess in a high probability. Gemp et al. (2024) subsequently developed LP-free methods based on cyclic projections and a Core Lagrangian formulation with stochastic updates. Both studies pursued more efficient least-core computation and compared its attributions with Shapley values in machine-learning applications. Nagarajan et al. (2026) identified a concrete source of advantage for the least core through *contextual distractors*: groups of features that help in less informative contexts but strongly harm important ones. They established conditions under which Shapley and other semivalues assign more credit to these groups than the least core, and demonstrated this effect in explaining biased image classifiers.

Our study addresses computational aspects of the least-core in this contextual-distractor setting—is the minimum-norm selector, the de facto standard selector, the most useful for retrieving contextual distractors, and how much does the sampling strategy of Yan & Procaccia (2021) affect retrieval?

[1] Further Shapley-related work is discussed in Section A.

As Gemp et al. (2024); Harris et al. (2026) focus on nonnegative payoffs and cannot assign negative payoffs to contextual distractors, their methods are outside the scope of this study.

3 PRELIMINARIES

Let $G = (N, v)$ be a coalitional game with player set $N = \{1, \dots, n\}$ and $v : 2^N \to \mathbb{R}$, $v(\emptyset) = 0$. An attribution is represented by a vector $x \in \mathbb{R}^n$, where x_i is the credit assigned to player i. We call such vectors *payoff vectors*. Write $x(S) := \sum_{i \in S} x_i$ for the total credit of a coalition S, and $e_S(x) := v(S) - x(S)$ for its *excess* (underpayment) relative to the worth $v(S)$. A payoff vector with $x(N) = v(N)$ is called *efficient*.

3.1 SHAPLEY VALUE AND LEAST CORE

We compare two payoff rules for the same coalitional game: the Shapley value averages marginal contributions, whereas the least core controls the worst coalition-level underpayment.

Definition 3.1 (Shapley value)**.** The Shapley value is the payoff vector x with

$$x_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S)). \tag{1}$$

In words, it averages the marginal contribution of player i over all contexts $S \subseteq N \setminus \{i\}$. In contrast, the least core focuses on the worst excess across coalitions. Throughout the paper, least-core constraints range over the proper coalition family $\mathcal{S} = 2^N \setminus \{\emptyset, N\}$, and we write $M = |\mathcal{S}| = 2^n - 2$. For efficient payoffs, both the empty and grand coalitions have zero excess. Excluding both allows the least-core value to be negative.

Definition 3.2 (ϵ-core)**.** For $\epsilon \in \mathbb{R}$, the ϵ-core is the set $\mathrm{Core}(G, \epsilon) \subset \mathbb{R}^n$ of payoffs that satisfy the following constraint:

$$x(S) \geq v(S) - \epsilon \ \text{ for all } S \in \mathcal{S}, \qquad x(N) = v(N). \tag{2}$$

The smallest ϵ with $\mathrm{Core}(G, \epsilon) \neq \emptyset$ is the *least-core value* ε^*. The set $\mathrm{LC}(\varepsilon^*) := \mathrm{Core}(G, \varepsilon^*)$ is the *least core*.

The least core is a set of payoff vectors, while a least-core payoff is one point in that set. The least-core value and one least-core payoff are computed by the linear program

$$\min_{x \in \mathbb{R}^n,\, \epsilon \in \mathbb{R}} \ \epsilon \quad \text{s.t.} \quad x(N) = v(N) \ \text{ and } \ x(S) \geq v(S) - \epsilon, \ \forall S \in \mathcal{S}. \tag{3}$$

Equivalently, it minimizes the maximum excess $\max_{S \in \mathcal{S}} e_S(x)$ over efficient payoffs. This worst-coalition criterion distinguishes the least core from the Shapley value.

3.2 SELECTORS FOR THE LEAST CORE

The least-core value ε^* is unique, but the minimizing payoff vector need not be unique; $x \in \mathrm{LC}(\varepsilon^*)$ is a least-core payoff. Without any specification, the returned payoff depends on the solver and the parameters. A *selector* is a rule that returns one least-core payoff from the least-core face. There are several natural selectors (Nagarajan et al., 2026). We describe two of them here.

Minimum-norm selector. The minimum-norm selector breaks ties by the ℓ_2 norm,

$$x_{\mathrm{mn}} := \sigma_{\mathrm{mn}}(\mathrm{LC}(\varepsilon^*)) := \operatorname*{arg\,min}_{x \in \mathrm{LC}(\varepsilon^*)} \|x\|_2^2. \tag{4}$$

The objective is strictly convex, so the minimizer is unique. Intuitively, among least-core payoffs with the same stability guarantee, this selector favors the point closest to the origin.

Nucleolus selector. The nucleolus[2] (Schmeidler, 1969; Maschler et al., 1979) refines the least core by minimizing the second-largest excess, then the third, and so on, without worsening the preceding minima. Fix an enumeration $S_1, \ldots, S_M$ of $\mathcal{S}$. For each candidate payoff x, let $(\sigma_1(x), \ldots, \sigma_M(x))$ be a permutation of $\{1, \ldots, M\}$ that sorts the excesses:

$$e_{S_{\sigma_1(x)}}(x) \geq e_{S_{\sigma_2(x)}}(x) \geq \cdots \geq e_{S_{\sigma_M(x)}}(x).$$

The nucleolus lexicographically minimizes this vector over efficient payoffs (cf. Section B.1).

The min-norm selector is intuitive and popular in literature (Yan & Procaccia, 2021; Harris et al., 2026; Nagarajan et al., 2026). The nucleolus is more aligned with the least-core idea as it optimizes the excess profile beyond the worst excess, while Yan & Procaccia (2021) showed that their sampling approach requires exponential queries to give a good approximation of the nucleolus. In Gemp et al. (2024), the returned payoff vector is determined implicitly by the iterative procedure and can depend on hyperparameters, initialization, sampling, and stopping. It is worth noting that their method restricts the payoff to the simplex, and thus cannot assign negative payoffs to contextual distractors.

3.3 Contextual distractors and a motivating example

We now introduce the contextual distractors and a motivating example that highlights the effect of the selector choice on the attribution.

Definition 3.3 (Contextual distractor (Nagarajan et al., 2026))**.** Let $D \subset N$ be a nonempty group. For a context $T \subseteq N \setminus D$, write $\Delta_D(T) = v(T \cup D) - v(T)$. For $\gamma > 0$, D is a γ-contextual distractor if $\Delta_D(T) \leq -\gamma$ for some such T.

Nagarajan et al. (2026) discussed when the least core and the Shapley value assign different attribution scores. A critical condition is that $\Delta_D(T) \leq -\gamma$ for $v(T) \geq \tau$ for some $\tau > 0$ and $\Delta_D(T) > 0$ otherwise. Namely, the contextual distractor D is harmful only when T is a (rare), high-value context, and otherwise harmless.

Throughout this paper, we focus on this case. The following game illustrates the contextual distractors D for useful players U and the importance of the selector choice.

A simple contextual-distractor game. Let U contain useful players with additive contributions m_i, and let D contain distractors:

$$v(S) = \sum_{i \in S \cap U} m_i + a|S \cap D| + b\,\mathbf{1}[U \subseteq S]\,\mathbf{1}[S \cap D = \emptyset], \tag{5}$$

where $\mathbf{1}[A]$ equals 1 if condition A holds and 0 otherwise, and $0 < a|D| < b$. Each distractor contributes $a > 0$ in ordinary coalitions, but destroys the bonus b when all useful players are present without any distractor. Formally, D is a contextual distractor with $\Delta_D(U) \leq a|D| - b$.

Consider four useful players with $m = (6, 8, 12, 14)$, two distractors, $a = 5$, and $b = 60$. Table 1 compares the payoff of the least core with different selectors. *LP* denotes the vertex payoff returned by the LP solver, which is determined by the solver and the parameters. The Shapley value and our method ELC are also given for reference. As can be seen, the Shapley value fails to identify the contextual distractors, assigning positive credit to them, but the least core methods correctly assign negative credit to at least one of them. Particularly, the LP with the min-norm and nucleolus selectors successfully assign negative credit to both distractors. The latter is more accurate in the positive credit assignment to the useful players; it retrieves m_i with an offset of 7.5 to each useful player. This common offset is consistent with the efficiency constraint $x(N) = v(N)$ and does not affect useful-player ordering.

4 Entropic least core

4.1 Definition and properties

We introduce entropic least core (ELC), which is a *smoothed* least-core payoff and approaches the least-core face in the limit of small temperature.

[2]Strictly, we consider the *prenucleolus*: we impose efficiency without individual rationality ($x_i \geq v(\{i\})$), allowing negative credit. This follows Yan & Procaccia (2021).

Table 1: Attribution in a six-player contextual-distractor game. Shapley assigns positive credit to both distractors. Least-core allocations assign negative total credit to the pair, but differ in individual signs and the relative credit of useful players.

Player	Shapley	LP	LP + min-norm	LP + nucleolus	ELC
useful 1 ($m_1 = 6$)	7	6	17.5	13.5	13.5
useful 2 ($m_2 = 8$)	9	8	17.5	15.5	15.5
useful 3 ($m_3 = 12$)	13	12	17.5	19.5	19.5
useful 4 ($m_4 = 14$)	15	44	17.5	21.5	21.5
distractor 1	3	5	-10	-10	-10
distractor 2	3	-25	-10	-10	-10
Max. excess	56	**30**	**30**	**30**	**30**

The least-core value minimizes the maximum excess over efficient payoffs:

$$\varepsilon^* = \min_{x:\, x(N)=v(N)} \hat{\varepsilon}(x), \qquad \text{where} \qquad \hat{\varepsilon}(x) := \max_{S\in\mathcal{S}} e_S(x). \tag{6}$$

ELC replaces this hard maximum by the log-sum-exp approximation

$$\varepsilon_\eta(x) = \eta \log \sum_{S\in\mathcal{S}} u_S^\eta(x), \qquad \text{where} \qquad u_S^\eta(x) := \exp\big(e_S(x)/\eta\big), \tag{7}$$

where $\eta > 0$ is the temperature parameter. Letting $M := |\mathcal{S}|$, the log-sum-exp bound gives

$$\hat{\varepsilon}(x) \le \varepsilon_\eta(x) \le \hat{\varepsilon}(x) + \eta \log M. \tag{8}$$

Thus $\varepsilon_\eta(x) \to \hat{\varepsilon}(x)$ as $\eta \to 0^+$.

Definition 4.1 (Entropic least core)**.** The η-entropic least core (η-ELC) is the payoff vector that minimizes the smoothed maximum excess over efficient payoff vectors:

$$x_\eta := \operatorname*{arg\,min}_{x:\, x(N)=v(N)} \varepsilon_\eta(x). \tag{9}$$

Entropic view. The replacement of the hard max by the log-sum-exp function has a natural interpretation of introducing an entropy regularization term. Let Δ be the simplex over the coalition family $\mathcal{S}$, and let $w = (w_S)_{S\in\mathcal{S}} \in \Delta$ be a reference distribution with full support. The variational representation of the log-sum-exp function suggests that for any $a \in \mathbb{R}^M$ and $\eta > 0$,

$$\max_{\lambda\in\Delta}\Big\{\sum_{S\in\mathcal{S}} \lambda_S a_S - \eta\, \mathrm{KL}(\lambda\|w)\Big\} = \eta \log \sum_{S\in\mathcal{S}} w_S\, e^{a_S/\eta}. \tag{10}$$

The left-hand side is a maximization over the simplex with regularization by the Kullback–Leibler divergence, called entropic regularization in the literature of optimal transport (Cuturi, 2013). The maximizer is unique and is the Gibbs distribution $\lambda_S \propto w_S e^{a_S/\eta}$. Setting $a_S = e_S(x)$ and $w_S = 1/M$ for all $S \in \mathcal{S}$ gives the ELC objective up to the additive constant $-\eta \log M$, giving the same minimizer.

Why not regularize the least core directly? A natural alternative is to compute the least-core polytope and then select a canonical point using a strongly convex or entropic regularizer. This does not remove the main computational bottleneck: the least core contains one constraint per coalition,

$$e_S(x) \le \varepsilon \qquad \forall S \in \mathcal{S},$$

and separation at a candidate (x, ε) requires solving $\max_{S\in\mathcal{S}} e_S(x)$. Efficient optimization or projection over such an exponentially described polytope therefore relies on an efficient separation oracle (Grötschel et al., 1988, pp. 102–132), while core and ε-core related problems are intractable for general compactly represented cooperative games without additional structure (Deng & Papadimitriou, 1994). Thus, regularizing the allocation changes how one selects among feasible least-core solutions, but leaves the hard coalition search intact. Entropic smoothing instead trades this combinatorial search for stochastic estimation, replacing the exact worst-coalition problem by sampling the influence of high-excess coalitions.

The following proposition characterizes the η-ELC and its limit, using the exponential-penalty result of Alvarez (2000) to establish convergence to the nucleolus.

Proposition 4.2 (ELC properties)**.** For $n \geq 2$ and $\mathcal{S} = 2^N \setminus \{\emptyset, N\}$, the following hold.

1. For every $\eta > 0$, the η-ELC is a singleton; namely, Equation (9) has a unique minimizer.
2. $0 \leq \hat{\varepsilon}(x_\eta) - \varepsilon^* \leq \eta \log M$. As $\eta \to 0^+$, $\varepsilon_\eta(x_\eta) \to \varepsilon^*$.
3. The exact ELC path converges to the nucleolus x_{nuc}; that is, $x_\eta \to x_{\text{nuc}}$ as $\eta \to 0^+$.

The proof is in Section B.3. The first two statements say that ELC is a principled smooth approximation to the least core: every positive temperature gives a unique payoff whose worst excess is within $\eta \log M$ of the least-core value. The third statement identifies the induced selector: it converges to the nucleolus, the classical lexicographic refinement of the least core.

Selector view. From this perspective, the KL-regularized objective is not only a smooth surrogate for the least-core value; it equips the approximate least core with a selector, inducing the unique payoff for each $\eta > 0$. As the approximation vanishes with $\eta \to 0^+$, this payoff smoothly converges to the nucleolus. Quadratic regularization instead selects the minimum-norm allocation in the vanishing-regularization limit (Li et al., 2026).

4.2 Computation

We compute ELC by projected gradient descent on the convex payoff objective over the efficiency hyperplane. Projection onto the efficiency hyperplane is closed form and requires no linear optimization oracle. Differentiating Equation (7) gives

$$\frac{\partial \varepsilon_\eta(x)}{\partial x_i} = -\frac{\sum_{S \in \mathcal{S},\, i \in S} u_S^\eta(x)}{\sum_{S \in \mathcal{S}} u_S^\eta(x)} = -\Pr_{S \sim \lambda_x^\eta}[\, i \in S\,], \tag{11}$$

where λ_x^η is the Gibbs distribution on $\mathcal{S}$,

$$\lambda_x^\eta(S) := \frac{u_S^\eta(x)}{\sum_{S' \in \mathcal{S}} u_{S'}^\eta(x)}. \tag{12}$$

Equation (11) shows that the gradient of the ELC objective is the negated membership probability under the Gibbs distribution λ_x^η, which puts more mass on coalitions with larger excess.

Let $r(x) \in \mathbb{R}^n$ with $r_i(x) = -\Pr_{S \sim \lambda_x^\eta}[i \in S]$. The projected gradient update is

$$x \leftarrow x - \alpha\big(r(x) - \bar{r}(x)\,\mathbf{1}\big), \tag{13}$$

where $\alpha > 0$ is the step size, $\bar{r}(x) := \frac{1}{n}\sum_{i=1}^n r_i(x)$, and $\mathbf{1}$ is the all-ones vector. Subtracting $\bar{r}(x)\mathbf{1}$ keeps $x(N) := \sum_{i=1}^n x_i$ unchanged over iterations. If we initialize, for example, at the uniform efficient payoff $x_0 = (v(N)/n)\mathbf{1}$, every iterate remains efficient. Standard smooth-convex guarantees give objective-error rates $O(1/T)$ after T iterations with step size $4\eta/n$ and $O(1/T^2)$ with Nesterov acceleration (Proposition C.2). In practice, we approximate the ELC path by annealing, computing η-ELC at successively smaller values of η.

5 Computation and Payoff Selection

This section presents the empirical validation of ELC. Throughout the paper, unless otherwise stated, all the experiments were run on a Linux server with an AMD EPYC 9124 16-core CPU and an NVIDIA RTX PRO 6000 Blackwell Max-Q with the following settings.

Experimental setups. **(ELC)** Our full-coalition experiments enumerate all $M = 2^n - 2$ nonempty proper coalitions and use uniform ELC weights $w_S = 1/M$. We optimize ELC by projected gradient descent with Nesterov acceleration, step size $\alpha = 4\eta/n$, and initial payoff $x_0 = (v(N)/n)\mathbf{1}$. We use nine annealing stages from $\eta = 1$ to 0.002, warm-starting each stage from the preceding payoff. Each stage stops at a projected-gradient infinity norm of 10^{-8} or after 10,000 updates. The full temperature schedule and solver details are in Section D. **(Baselines)** We compare ELC with LP, LP + min-norm, and LP + nucleolus, using HiGHS for linear programs and SLSQP for minimum-norm selection. The nucleolus solver follows the successive-LP procedure in Section B.1.

Table 3: ELC approximates the nucleolus at lower computational cost. Distances are $d_{\text{nuc}} = \|x_{\text{ELC}} - x_{\text{nuc}}\|_2$ and $d_{\text{mn}} = \|x_{\text{ELC}} - x_{\text{mn}}\|_2$; gap is $\hat{\varepsilon}(x_{\text{ELC}}) - \varepsilon^*$. Distances and gaps use medians across 20 games per random setting or ten UCI splits, after taking within-game medians over three repetitions. Times average optimization runtimes, excluding game construction and independent evaluation. Further details are in Section F.1. † One failed min-norm reference is excluded. ‡ House nucleolus time averages 15 completed repetitions, excluding 15 timeouts under a 60-minute limit; distances use all ten splits.

	ELC (CPU)			Reference	
Setting (n)	Time (s)	d_{nuc} (10^{-3})	Gap (10^{-3})	d_{mn} (10^{-3})	LP + nucleolus (s)
Random (6)	0.253	1.97	0.59	98.28	1.117
Random (7)	0.190	1.66	0.79	24.38	1.065
Random (8)	0.384	2.84	0.28	81.36	1.369
Medical (9)	0.089	0.90	0.10	4.56	1.041
Chemical (13)	1.146	9.64	1.02	21.59†	45.321
House (16)	5.414	8.96	1.27	13.18	2150.567‡

Least core value precision. Defining random games with $v(S) = |S|/n + \xi_S$, independent $\xi_S \sim \mathcal{N}(0, 0.3^2)$, and $v(N) = 1$, ELC attains mean hard-excess gaps $\hat{\varepsilon}(x) - \varepsilon^*$ of order 10^{-3} against the full LP (Table 2).

Table 2: Least-core quality. Entries are mean ± standard error over five games. Rel. gap uses the coalition-value range.

n	M	Gap (10^{-3})	Rel. gap (10^{-4})
8	254	0.81 ± 0.17	4.00 ± 0.81
10	1,022	0.96 ± 0.30	4.20 ± 1.32
12	4,094	1.23 ± 0.12	4.96 ± 0.50
14	16,382	1.34 ± 0.14	5.19 ± 0.51
16	65,534	1.17 ± 0.10	4.17 ± 0.35
18	262,142	1.64 ± 0.15	5.72 ± 0.62

ELC payoff in the limit. We introduce UCI games from Yan & Procaccia (2021) in addition to random games, and observe in Table 3 that the ELC payoff approaches the nucleolus in the limit of $\eta \to 0$. As temperature decreases, the median distance to the nucleolus falls monotonically in all three settings (Figure 1); individual paths need not be monotone.

Runtime. Table 3 compares finite-annealing ELC with the successive-LP nucleolus solver at their attained accuracies. ELC takes less solver time in the reported settings. Table 7 further shows that GPU acceleration benefits ELC at larger coalition counts. Sampling reduces the cost of the initial LP, but the subsequent min-norm selection can be slower and less reliable. The attribution results in Section 6 show that sampling particularly degrades distractor identification, while useful-player identification remains comparatively strong.

6 Identifying Useful and Harmful Players

To test how context rarity and constraint sampling affect identification, we generalize Equation (5).

Contextual-distractor game. Let $U, D \subseteq N$ be disjoint sets of useful players and distractors; the remaining players are neutral. Useful player i contributes $m_i > 0$, and distractor j contributes $a_j > 0$. A bonus $b > 0$ becomes available when at least τ useful players are present; each included distractor reduces it by $h_j > 0$. For a coalition $S \subseteq N$, define

$$v(S) = \sum_{i \in S \cap U} m_i + \sum_{j \in S \cap D} a_j + \mathbf{1}[|S \cap U| \geq \tau] \left(b - \sum_{j \in S \cap D} h_j \right)_+, \tag{14}$$

where $(z)_+ := \max\{z, 0\}$. Setting $\tau = |U|$, $a_j = a$, and $h_j = b$ recovers Equation (5). Adding distractor j to a coalition with at least τ useful players and no distractors reduces its value by $h_j - a_j$ when $a_j < h_j \leq b$.

Setup. We use $(n, |U|, |D|) = (10, 6, 2)$ and $(14, 8, 3)$, with neutral players filling the remaining positions. For each of ten game seeds, $\tau \in \{\lceil |U|/2 \rceil, \lceil 3|U|/4 \rceil, |U|\}$; $b = 2|U|$ and $a_j = 0.1|U|$.

Table 4: Full-coalition ELC identifies and orders both player groups; sampling mainly reduces distractor identification. AP^+/AP^- measure useful/distractor identification. Negative rate is the fraction of distractors receiving negative credit; order accuracy is the fraction of distinct-strength pairs ranked correctly, counting ties as incorrect. All scores range from 0 to 1; higher is better, and bold marks the best mean within each budget, including ties. Means average samples within each of 60 games, then across games. Sampled methods share coalition observations. Further evaluation details are in Section G.

Constraints	Method	AP^+	AP^-	Distractor negative rate	Useful order accuracy	Harm order accuracy
Full	Shapley	**1.000**	0.794	0.639	**1.000**	**1.000**
	LP	**1.000**	0.826	0.667	0.918	0.617
	LP + min-norm	**1.000**	**1.000**	**1.000**	0.667	0.333
	ELC	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
50% sample	Shapley	0.920	0.689	0.635	0.563	0.740
	LP	0.998	0.827	0.693	**0.922**	0.636
	LP + min-norm	0.999	**0.967**	**0.962**	0.667	0.432
	ELC	**1.000**	**0.967**	**0.962**	0.899	**0.959**
10% sample	Shapley	0.883	0.546	0.474	0.545	0.689
	LP	0.980	0.824	0.770	**0.760**	0.647
	LP + min-norm	**0.983**	**0.879**	**0.877**	0.725	0.506
	ELC	**0.983**	0.860	0.864	0.720	**0.842**

Useful effects vary across seeds. Distractors have distinct, equally spaced h_j/b values from 0.25 to 0.75, permuted across players. Each game has five nested 10%/50% uniform samples, shared across LP, LP + min-norm, and ELC. All methods are also run on the full family. We evaluate average precision (AP): AP^+ ranks x_i against useful-player membership, and AP^- ranks $-x_i$ against distractor membership. We also measure payoff signs and within-group strength ordering.

Results. Table 4 summarizes the results. First, with full coalitions, the AP^+ and AP^- scores show that all methods identify all useful players correctly, but the Shapley value and the least core without a specified selector show limited performance in identifying distractors. Notably, ELC not only identifies the distractor players correctly, but also ranks them more accurately than the minimum-norm selector. In the 50%/10% sampling settings, AP^- deteriorates across methods except LP, while AP^+ remains strong. This suggests that the sampling effect is more pronounced on distractor identification than on useful-player identification. Namely, while the least core with a selector is more advantageous than the Shapley value for identifying the distractors as Nagarajan et al. (2026) pointed out, the sampling approach, a typical acceleration strategy in the literature (Yan & Procaccia, 2021; Gemp et al., 2024), can reduce this advantage. The loss is more pronounced at higher context thresholds and with fewer sampled coalitions (Figures 2 and 3).

We observed that the larger drop in AP^- than in AP^+ persists for ELC and LP + min-norm when $|U| = |D|$, so unequal group sizes alone do not explain this difference (see Table 9).

7 Applications to Training-Data Valuation

We now compare the Shapley value and the least core methods in downstream tasks. We adopt the training-data valuation task from Yan & Procaccia (2021). Further results and experiments are provided in Sections H and I.

Metrics. Training points were removed in increments of 5%, refitting the predictor after removal. Removing the highest-ranked points (Best) tests whether important examples receive high credit; lower retained accuracy is better. Removing the lowest-ranked points (Worst) tests whether useful examples are retained; higher is better. We integrate each accuracy curve over 0–35% removal and divide by 0.35 to obtain the normalized area under the curve (AUC). Differences are paired within repetition; uncertainty is computed over repetitions, not over points on the removal curve.

Table 5: Paired differences in normalized removal AUC (percentage points): ELC minus each baseline. Best removes the highest-ranked training points; Worst removes the lowest-ranked. Negative differences favor ELC for Best; positive differences favor ELC for Worst. CI denotes the 95% bootstrap confidence interval for the mean difference. N_{query} is the coalition-query budget.

		ELC − min-norm LC		ELC − Shapley	
N_{query}	Removal	Mean	95% CI	Mean	95% CI
Gaussian (20 repetitions per budget)					
5K	Best ↓	−0.463	[−0.613, −0.306]	−3.312	[−3.889, −2.769]
5K	Worst ↑	+0.118	[+0.037, +0.197]	+0.076	[−0.206, +0.330]
10K	Best ↓	−0.439	[−0.582, −0.305]	−4.646	[−5.330, −4.023]
10K	Worst ↑	+0.121	[+0.046, +0.193]	+0.468	[+0.097, +0.831]
25K	Best ↓	−0.355	[−0.489, −0.217]	−4.292	[−4.812, −3.779]
25K	Worst ↑	+0.210	[+0.146, +0.272]	+0.728	[+0.446, +1.005]
50K	Best ↓	−0.495	[−0.673, −0.327]	−3.185	[−3.609, −2.715]
50K	Worst ↑	+0.165	[+0.083, +0.255]	+0.382	[+0.167, +0.608]
Dog–fish (5 repetitions per budget)					
5K	Best ↓	−0.074	[−0.143, −0.005]	−0.402	[−0.581, −0.217]
5K	Worst ↑	+0.012	[−0.002, +0.024]	+0.005	[−0.057, +0.086]
10K	Best ↓	+0.005	[−0.036, +0.045]	−0.360	[−0.500, −0.219]
10K	Worst ↑	+0.033	[+0.000, +0.071]	+0.043	[+0.029, +0.064]
25K	Best ↓	−0.095	[−0.195, −0.007]	−0.593	[−0.671, −0.512]
25K	Worst ↑	+0.024	[+0.005, +0.048]	+0.088	[+0.019, +0.164]
50K	Best ↓	−0.052	[−0.086, −0.019]	−0.345	[−0.414, −0.283]
50K	Worst ↑	+0.036	[−0.024, +0.088]	+0.326	[+0.200, +0.486]

Results. Table 5 shows that ELC improves mean removal AUC over the minimum-norm least core and Shapley value in nearly all comparisons. ELC-minus-baseline AUC differences are negative for best-first removal and positive for worst-first removal, except for Dogfish best-first removal at 10K queries versus the minimum-norm least core.

8 Conclusion

We studied least-core attribution for contextual distractors, which cause substantial harm in rare, high-value contexts, from a computational perspective. Our experiments show that selector choice is crucial for distractor retrieval and that coalition sampling degrades distractor identification more severely than useful-player identification. We introduced entropic least core (ELC), a smooth log-sum-exp approximation whose minimizer converges to the nucleolus as the temperature $\eta \to 0$. In our experiments, ELC approximates the nucleolus in less solver time than the successive-LP procedure. ELC and the minimum-norm selector both achieve high average precision for distractor identification in the tested games; ELC further ranks distractors more accurately by harm.

Limitations and future work. Least-core attribution for model explanation is still developing, with two open questions. First, the practical advantages of the least core over the Shapley value remain incompletely understood. Existing experiments span feature attribution, data valuation, and label correction, but these are general tasks. Identifying tasks that benefit specifically from contextual-distractor retrieval remains an important direction. Second, full coalition enumeration has exponential cost in the number of players. Recent work proposes approximation methods, but the method of Gemp et al. (2024) restricts payoffs to be nonnegative, and coalition sampling as in Yan & Procaccia (2021) can degrade distractor identification. Designing a sampling method, or other approximation techniques, that preserves this ability while providing approximation guarantees is an open direction for scalable least-core computation.

AI USE STATEMENT

Generative AI tools assisted with manuscript drafting and editing, mathematical discussion, experiment implementation, result auditing, and figure preparation. Reported experimental values come from the saved numerical experiments. The authors are responsible for the final manuscript, code, and interpretation of the results.

REPRODUCIBILITY STATEMENT

Proofs of the variational representation and smoothness are in Sections B.2 and C. Solver details, game families, and evaluation metrics are in Sections 5 to G. The accompanying repository contains the experimental code, saved results, and instructions for reproducing each experiment and display.

REFERENCES

Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298:103502, 2021.

Jacob A. Abernethy, Robert E. Schapire, and Umar Syed. Lexicographic optimization: Algorithms and stability. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 2503–2511, 2024. URL https://proceedings.mlr.press/v238/abernethy24a.html.

F. Alvarez. Absolute minimizer in convex programming by exponential penalty. *Journal of Convex Analysis*, 7(1):197–202, 2000.

Stefan Blücher, Johanna Vielhaben, and Nils Strodthoff. PredDiff: Explanations and interactions from conditional expectations. *Artificial Intelligence*, 312:103774, 2022. doi: 10.1016/j.artint.2022.103774.

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.

Wonjoon Chang, Myeongjin Lee, and Jaesik Choi. Rethinking Shapley value for negative interactions in non-convex games. In *International Conference on Learning Representations (ICLR)*, 2025.

Ian Covert, Scott M. Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/c7bf0b7c1a86d5eb3be2c722cf2cf746-Abstract.html.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, pp. 2292–2300. Curran Associates Inc., 2013. URL https://papers.neurips.cc/paper_files/paper/2013/hash/af21d0c97db2e27e13572cbf59eb343d-Abstract.html.

Xiaotie Deng and Christos H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994. doi: 10.1287/moor.19.2.257.

P. Dubey, A. Neyman, and R. J. Weber. Value theory without efficiency. *Mathematics of Operations Research*, 6(1):122–128, 1981. doi: 10.1287/moor.6.1.122.

Ian Gemp, Marc Lanctot, Luke Marris, Yiran Mao, Edgar Duéñez-Guzmán, Sarah Perrin, Andras Gyorgy, Romuald Elie, Georgios Piliouras, Michael Kaisers, Daniel Hennes, Kalesha Bullard, Kate Larson, and Yoram Bachrach. Approximating the core via iterative coalition sampling. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 669–678, 2024. URL https://www.ifaamas.org/Proceedings/aamas2024/pdfs/p669.pdf.

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

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory*, 28:547–565, 1999. doi: 10.1007/s001820050125.

Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, Heidelberg, 1988. doi: 10.1007/978-3-642-97881-4.

Keegan Harris, Siddharth Prasad, and Asher Trockman. In-context credit assignment via the core. *arXiv preprint arXiv:2605.06920*, 2026. URL https://arxiv.org/abs/2605.06920.

Neil Jethani, Mukund Sudarshan, Ian Covert, Su-In Lee, and Rajesh Ranganath. FastSHAP: Real-time Shapley value estimation. In *International Conference on Learning Representations (ICLR)*, 2022.

Yongchan Kwon and James Zou. Beta Shapley: a unified and noise-reduced data valuation framework for machine learning. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 8780–8802. PMLR, 2022. URL https://proceedings.mlr.press/v151/kwon22a.html.

Weida Li, Zhuanghua Liu, Yaoliang Yu, and Bryan Kian Hsiang Low. Nonlinear axiomatic attribution for cooperative games. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 3581–3603. PMLR, 2026. URL https://proceedings.mlr.press/v337/li26c.html.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

M. Maschler, B. Peleg, and L. S. Shapley. Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of Operations Research*, 4(4):303–338, 1979. doi: 10.1287/moor.4.4.303.

Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. Sampling permutations for Shapley value estimation. *Journal of Machine Learning Research*, 23(43):1–46, 2022.

S. G. Nagarajan, T. Yamauchi, and H. Kera. Seeing through distractions: Stable attribution via the core. In *NExT-Game Workshop at ICML*, 2026.

Y. Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, 2 edition, 2018. doi: 10.1007/978-3-319-91578-4.

D. Schmeidler. The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17(6):1163–1170, 1969. doi: 10.1137/0117107.

L. S. Shapley. A value for n-person games. In *Contributions to the Theory of Games II*. Princeton University Press, 1953.

Erik Štrumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(1):1–18, 2010. URL https://www.jmlr.org/papers/v11/strumbelj10a.html.

Kosuke Sumiyasu, Kazuhiko Kawamoto, and Hiroshi Kera. Game-theoretic understanding of misclassification. arXiv preprint arXiv:2210.03349, 2022.

Kosuke Sumiyasu, Kazuhiko Kawamoto, and Hiroshi Kera. Identifying important group of pixels using interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6017–6026, 2024.

Mukund Sundararajan and Amir Najmi. The many Shapley values for model explanation. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9269–9278. PMLR, 2020. URL https://proceedings.mlr.press/v119/sundararajan20b.html.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Jiachen T. Wang and Ruoxi Jia. Data Banzhaf: A robust data valuation framework for machine learning. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 6388–6421. PMLR, 2023. URL https://proceedings.mlr.press/v206/wang23e.html.

Toshinori Yamauchi, Hiroshi Kera, and Kazuhiko Kawamoto. Explaining object detectors via collective contribution of pixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17046–17056, 2026.

T. Yan and A. D. Procaccia. If you like Shapley then you'll love the core. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 5751–5759, 2021. doi: 10.1609/aaai.v35i6.16721.

A Additional related work

We expand the attribution literature discussed in Section 2.

The Shapley value (Shapley, 1953) assigns credit by averaging a player's marginal contribution over the orders in which players can join a coalition. It supports local feature explanations in Štrumbelj & Kononenko (2010) and SHAP (Lundberg & Lee, 2017), global feature importance in SAGE (Covert et al., 2020), and the valuation of training examples in Data Shapley (Ghorbani & Zou, 2019). Related game-theoretic approaches also study groups of players and their interactions, including in computer-vision explanations (Grabisch & Roubens, 1999; Sumiyasu et al., 2022; 2024; Blücher et al., 2022; Chang et al., 2025). The choice of players and coalition values determines what an attribution explains. Even for feature attribution, how absent features are represented affects the explanation (Sundararajan & Najmi, 2020); Aas et al. (2021) specifically address dependencies between features.

Computational work reduces the cost of Shapley attribution through permutation sampling (Castro et al., 2009; Mitchell et al., 2022) or learned estimators such as FastSHAP (Jethani et al., 2022). These methods approximate an averaging-based allocation rule. Semivalues generalize this rule by changing the weighting of coalition sizes (Dubey et al., 1981; Kwon & Zou, 2022; Wang & Jia, 2023). Such averaging is natural when the target is average contribution, but it can obscure harm concentrated in important contexts, as formalized by Nagarajan et al. (2026). The least core instead minimizes the largest coalition shortfall. This distinction concerns which allocation is sought, separately from how accurately it is computed: our comparisons examine both the allocation rule and the effect of coalition sampling on the resulting explanation.

B Least-core selectors and entropic regularization

This section supplies the selector definitions and proofs used in Sections 3 and 4. We describe the nucleolus, establish the entropic variational representation, and apply Alvarez (2000) to identify the ELC limit.

B.1 Nucleolus as a lexicographic least-core selector

The nucleolus refines the least core by minimizing coalition excesses lexicographically. For an efficient payoff x, recall the excess of a coalition S:

$$e_S(x) := v(S) - x(S).$$

Large $e_S(x)$ means that coalition S is underpaid by x. Fix an enumeration $S_1, \dots, S_M$ of $\mathcal{S}$. For each payoff x, let $(\sigma_1(x), \dots, \sigma_M(x))$ be a permutation of $\{1, \dots, M\}$ such that

$$e_{S_{\sigma_1(x)}}(x) \geq e_{S_{\sigma_2(x)}}(x) \geq \cdots \geq e_{S_{\sigma_M(x)}}(x).$$

Here $\sigma_k(x)$ is the index of the coalition with the kth-largest excess at x. The coalitions $S_1, \dots, S_M$ are fixed, while the sorting permutation depends on x. Ties may be ordered arbitrarily, since they do not change the sorted vector. The nucleolus is the efficient payoff whose sorted deficit vector is lexicographically minimal:

$$x_{\text{nuc}} := \operatorname*{lex\,arg\,min}_{x:\, x(N)=v(N)} \left(e_{S_{\sigma_1(x)}}(x), e_{S_{\sigma_2(x)}}(x), \dots, e_{S_{\sigma_M(x)}}(x)\right).$$

The first coordinate of this lexicographic program is the least-core value ε^*. Thus the nucleolus lies in the least core, and can be read as a selector for $\mathrm{LC}(\varepsilon^*)$. It is stricter than picking an arbitrary least-core point: after minimizing the worst deficit, it minimizes the next-worst remaining deficit, then the next one, until a single payoff remains.

The finite procedure is a sequence of linear programs. At level k, some coalitions have already been fixed at previous deficit levels $\theta_1, \dots, \theta_{k-1}$. Among all payoffs satisfying those equalities, solve

$$\begin{aligned} \min_{x,\theta_k} \quad & \theta_k \\ \text{s.t.} \quad & x(N) = v(N), \\ & e_S(x) = \theta_j \quad \text{for all coalitions fixed at level } j < k, \\ & e_S(x) \leq \theta_k \quad \text{for every unfixed coalition } S. \end{aligned} \tag{NUC-k}$$

A coalition is fixed only if its excess equals the current optimum throughout the optimal face. Tightness at one LP solution is insufficient: a constraint may be tight at the returned vertex but slack at another optimal payoff. The following procedure tests this by maximizing each coalition's slack over the face.

Let F contain the fixed coalitions and $\theta(S)$ their fixed excess levels:

Input: $\{(S, v(S)) : S \in \mathcal{S}\},\ v(N)$.
Initialize: $F \leftarrow \emptyset,\ k \leftarrow 1$.
Repeat: solve (NUC-k), obtaining its optimal value θ_k.
$T \leftarrow \emptyset$.
For each $S \in \mathcal{S} \setminus F$:
maximize $\theta_k - e_S(x)$ over the optimal face of (NUC-k).
If the maximum is $0,$ add S to T.
Set $F \leftarrow F \cup T$ and $\theta(S) \leftarrow \theta_k$ for $S \in T$.
$k \leftarrow k + 1$.
Until: x is uniquely determined.
Output: x.

B.2 Entropic representation

The following identity gives the entropic interpretation of ELC and its approximation bound for the maximum excess.

Proposition B.1 (Entropic variational representation)**.** Let Δ be the probability simplex on $\mathcal{S}$, let $w \in \Delta$ have full support, and let $a \in \mathbb{R}^M$ and $\eta > 0$. Then

$$\max_{\lambda \in \Delta} \left\{ \sum_S \lambda_S a_S - \eta \mathrm{KL}(\lambda \| w) \right\} = \eta \log \sum_S w_S e^{a_S/\eta}.$$

The unique maximizer is $\lambda^*_S = w_S e^{a_S/\eta} / \sum_T w_T e^{a_T/\eta}$. For $a_S = e_S(x)$ and $w_S = 1/M$, the value equals $\varepsilon_\eta(x) - \eta \log M$, and

$$\hat{\varepsilon}(x) \le \varepsilon_\eta(x) \le \hat{\varepsilon}(x) + \eta \log M, \qquad \hat{\varepsilon}(x) := \max_{S \in \mathcal{S}} e_S(x).$$

Proof. Set $Z = \sum_S w_S e^{a_S/\eta}$. Substituting $\log \lambda^*_S = \log w_S + a_S/\eta - \log Z$ gives

$$\sum_S \lambda_S a_S - \eta \mathrm{KL}(\lambda \| w) = \eta \log Z - \eta \mathrm{KL}(\lambda \| \lambda^*).$$

The KL divergence is nonnegative and vanishes only at $\lambda = \lambda^*$, proving the identity and uniqueness. For uniform w, evaluating the variational objective at a point mass on a maximizing coalition gives the lower bound $\hat{\varepsilon}(x) - \eta \log M$; its upper bound is $\hat{\varepsilon}(x)$ because $\sum_S \lambda_S e_S(x) \le \hat{\varepsilon}(x)$ and $\mathrm{KL}(\lambda \| w) \ge 0$. Adding $\eta \log M$ proves the stated bounds. □

B.3 ELC as an exponential-penalty trajectory

We prove Proposition 4.2, including convergence to the nucleolus, by expressing ELC as an exponential-penalty trajectory. We first establish the compactness needed for existence and for the penalty theorem.

Lemma B.2 (Compactness of excess sublevel sets)**.** For $n \ge 2$ and $\mathcal{S} = 2^N \setminus \{\emptyset, N\}$, let $\mathcal{X} = \{x \in \mathbb{R}^n : x(N) = v(N)\}$. For every $B \in \mathbb{R}$, the set

$$K_B := \{x \in \mathcal{X} : \hat{\varepsilon}(x) \le B\}$$

is compact, where $\hat{\varepsilon}(x) := \max_{S \in \mathcal{S}} e_S(x)$. In particular, the least core is nonempty and compact.

Proof. Membership in K_B means that $v(S) - x(S) \le B$ for every $S \in \mathcal{S}$. Taking $S = \{i\}$ gives

$$v(\{i\}) - x_i \le B, \qquad \text{so} \qquad x_i \ge v(\{i\}) - B.$$

Taking $S = N \setminus \{i\}$ and using $x(N \setminus \{i\}) = v(N) - x_i$ gives

$$v(N \setminus \{i\}) - v(N) + x_i \le B, \qquad \text{so} \qquad x_i \le B + v(N) - v(N \setminus \{i\}).$$

Thus every coordinate has a finite lower and upper bound, and K_B is bounded. It is also closed, since $\hat{\varepsilon}$ is continuous.

Choose any $\bar{x} \in \mathcal{X}$ and fix $B = \hat{\varepsilon}(\bar{x})$. Then K_B is nonempty and compact, so $\hat{\varepsilon}$ attains a minimum there. Outside K_B, $\hat{\varepsilon}(x) > B$, so this is also a minimum on $\mathcal{X}$. The least core is therefore nonempty; it is a closed subset of this compact K_B, and hence is compact. □

We use Proposition B.1 to prove existence and approximation quality, then apply the exponential-penalty theorem of Alvarez (2000) to identify the limiting selector.

Proposition 4.2 (ELC properties)**.** For $n \ge 2$ and $\mathcal{S} = 2^N \setminus \{\emptyset, N\}$, the following hold.

1. For every $\eta > 0$, the η-ELC is a singleton; namely, Equation (9) has a unique minimizer.
2. $0 \le \hat{\varepsilon}(x_\eta) - \varepsilon^* \le \eta \log M$. As $\eta \to 0^+$, $\varepsilon_\eta(x_\eta) \to \varepsilon^*$.
3. The exact ELC path converges to the nucleolus x_{nuc}; that is, $x_\eta \to x_{\text{nuc}}$ as $\eta \to 0^+$.

Proof of Proposition 4.2. **Existence and uniqueness (statement 1).** Let $\mathcal{X} := \{x \in \mathbb{R}^n : x(N) = v(N)\}$ be the efficient-payoff hyperplane, and choose any $\bar{x} \in \mathcal{X}$. Recall that $e_S(x) := v(S) - x(S)$. The ELC objective is

$$\varepsilon_\eta(x) := \eta \log \sum_{S \in \mathcal{S}} \exp\left(\frac{e_S(x)}{\eta}\right).$$

Consider

$$L_\eta := \{x \in \mathcal{X} : \varepsilon_\eta(x) \le \varepsilon_\eta(\bar{x})\}.$$

This set contains $\bar{x}$ and is closed by continuity.

Write $\hat{\varepsilon}(x) := \max_{S \in \mathcal{S}} e_S(x)$. By the log-sum-exp bound in Proposition B.1, it satisfies $\hat{\varepsilon}(x) \le \varepsilon_\eta(x)$. Hence $L_\eta \subseteq K_{\varepsilon_\eta(\bar{x})}$, where $K_B := \{x \in \mathcal{X} : \hat{\varepsilon}(x) \le B\}$ is compact by Lemma B.2. The continuous ELC objective attains its minimum on the compact set L_η, and points outside this set cannot improve that minimum. By Lemma C.1, ε_η is strictly convex on $\mathcal{X}$, so its minimizer x_η is unique.

Approximation quality (statement 2). Write $\varepsilon^* := \min_{x \in \mathcal{X}} \hat{\varepsilon}(x)$ for the least-core value, and choose a payoff x^0 attaining this minimum. Such a payoff exists by Lemma B.2. Let $M := |\mathcal{S}|$. Then Proposition B.1 and optimality of x_η give

$$\varepsilon^* \le \hat{\varepsilon}(x_\eta) \le \varepsilon_\eta(x_\eta) \le \varepsilon_\eta(x^0) \le \varepsilon^* + \eta \log M.$$

The first inequality holds because ε^* is the minimum of $\hat{\varepsilon}$ on $\mathcal{X}$. The second is the lower log-sum-exp bound. The third follows because x_η minimizes ε_η, and the last applies the upper log-sum-exp bound to x^0, for which $\hat{\varepsilon}(x^0) = \varepsilon^*$. This proves the hard-gap bound and convergence of the smooth value.

Identifying the selector (statement 3). We use the following affine case of Alvarez (2000, Theorem 3.1). Let $f_0, f_1, \ldots, f_m : \mathbb{R}^d \to \mathbb{R}$ be affine functions, and let Z^* be the set of solutions to

$$\min_{z \in \mathbb{R}^d} f_0(z) \quad \text{s.t.} \quad f_i(z) \le 0 \quad (i = 1, \ldots, m).$$

If Z^* is nonempty and compact, then, for every $\eta > 0$, the function

$$f_0(z) + \eta \sum_{i=1}^{m} \exp\big(f_i(z)/\eta\big)$$

has a unique minimizer. As η tends to zero through positive values, this minimizer converges to the point in Z^* that lexicographically minimizes the constraint values $f_1(z), \ldots, f_m(z)$ sorted from largest to smallest. Alvarez calls this point the *absolute minimizer*.

To apply this result, introduce a scalar t that upper-bounds every excess and write the least-core problem as

$$P: \quad \min_{x\in\mathcal{X},\, t\in\mathbb{R}} t \quad \text{s.t.} \quad e_S(x) - t \le 0 \quad (S \in \mathcal{S}).$$

We first check the affine-function assumption. Eliminate efficiency by writing $x = \bar{x} + Qy$, where $y \in \mathbb{R}^{n-1}$ and the columns of $Q \in \mathbb{R}^{n\times(n-1)}$ form a basis for $\{u \in \mathbb{R}^n : \sum_i u_i = 0\}$. The functions $f_0(y,t) = t$ and $f_S(y,t) = e_S(\bar{x} + Qy) - t$, one for each $S \in \mathcal{S}$, are affine in the joint variable (y,t).

Next, the optimal set of P is $S(P) = \{(x,\varepsilon^*) : x \in \mathcal{X}, \hat{\varepsilon}(x) = \varepsilon^*\}$, which is nonempty and compact by Lemma B.2. The change of coordinates is a continuous bijection with continuous inverse, so the optimal set in (y,t) is also nonempty and compact. Both assumptions of the stated affine case are therefore satisfied.

It remains to identify ELC with the exponential penalty in the theorem. For P, the penalized problem is

$$\min_{x\in\mathcal{X},\, t\in\mathbb{R}} \Phi_\eta(x,t), \qquad \Phi_\eta(x,t) := t + \eta \sum_{S\in\mathcal{S}} \exp\big((e_S(x) - t)/\eta\big). \tag{15}$$

For fixed x, this function is strictly convex in t, and its stationarity condition is $\sum_{S\in\mathcal{S}} \exp((e_S(x) - t)/\eta) = 1$. Therefore its minimizer is $t = \varepsilon_\eta(x)$, and $\min_t \Phi_\eta(x,t) = \varepsilon_\eta(x) + \eta$.

The joint minimizer is consequently $(x_\eta, \varepsilon_\eta(x_\eta))$. The theorem now gives convergence of this pair to the absolute minimizer on $S(P)$ as η tends to zero through positive values.

Finally, on $S(P)$ the constraint values are $e_S(x) - \varepsilon^*$. Subtracting the same constant ε^* from every excess leaves their lexicographic order unchanged. Restricting to the least core has already minimized the largest excess, so lexicographically minimizing the remaining excesses gives the nucleolus, as defined in Section B.1. The absolute minimizer's payoff component is thus the nucleolus x_{nuc}. Hence $x_\eta \to x_{\text{nuc}}$. □

Relation to Abernethy et al. (2024). Abernethy et al. (2024) also study lexicographic optimization through exponential-loss minimization. Our convergence result for the exact ELC path, however, follows from the earlier exponential-penalty theorem of Alvarez (2000). Abernethy et al. (2024) additionally establish convergence guarantees under lexicographic stability, including for approximate minimizers with controlled errors, and derive convergence-rate bounds; we do not use these results. For optimization, they analyze a multiplicative-weights algorithm that, with their choice of learning rates, is equivalent to Frank–Wolfe, whereas we use projected gradient descent directly in payoff space under the efficiency constraint (Section 4.2).

C Smoothness and Optimization Rates

This section establishes the curvature and convergence bounds used by the projected-gradient method in Section 4.

C.1 Smoothness of the ELC Objective

We bound the Hessian to justify the step size $4\eta/n$ and show strict convexity on the efficiency hyperplane, which ensures uniqueness.

Lemma C.1 (Smoothness of the ELC objective)**.** The function ε_η is convex and β-smooth with $\beta = n/(4\eta)$, meaning that its gradient is β-Lipschitz in the Euclidean norm. On the efficiency hyperplane it is strictly convex.

Proof. The gradient and Hessian of ε_η are

$$\nabla_x \varepsilon_\eta(x) = -\mathbb{E}_{S\sim\lambda_x^\eta}[\mathbf{1}_S], \tag{16}$$

$$\nabla_x^2 \varepsilon_\eta(x) = \frac{1}{\eta} \operatorname{Cov}_{S\sim\lambda_x^\eta}(\mathbf{1}_S) \succeq 0, \tag{17}$$

where $(\mathbf{1}_S)_i = \mathbf{1}[i \in S]$ is the coalition-membership vector, $\mathbb{E}$ denotes expectation, Cov covariance, and $\succeq 0$ positive semidefiniteness. Convexity follows.

Each coordinate of $\mathbf{1}_S$ is Bernoulli: writing Var for variance and $m_i(x) := \Pr_{S\sim\lambda_x^\eta}[i \in S]$,

$$\mathrm{Var}_{S\sim\lambda_x^\eta}\big((\mathbf{1}_S)_i\big) = m_i(x)\big(1 - m_i(x)\big) \le \tfrac{1}{4}.$$

Writing $\|\cdot\|_{\mathrm{op}}$ for the Euclidean operator norm and tr for the trace, the positive semidefinite Hessian's operator norm equals its largest eigenvalue and is at most its trace:

$$\big\|\nabla_x^2 \varepsilon_\eta(x)\big\|_{\mathrm{op}} = \frac{1}{\eta}\,\big\|\mathrm{Cov}_{S\sim\lambda_x^\eta}(\mathbf{1}_S)\big\|_{\mathrm{op}} \le \frac{1}{\eta}\,\mathrm{tr}\big(\mathrm{Cov}_{S\sim\lambda_x^\eta}(\mathbf{1}_S)\big) \le \frac{n}{4\eta}. \tag{18}$$

Hence $\nabla\varepsilon_\eta$ is β-Lipschitz with $\beta = n/(4\eta)$.

For strict convexity on the efficiency hyperplane, take a nonzero $d \in \mathbb{R}^n$ with $\mathbf{1}^\top d = 0$. The singleton coalitions have positive Gibbs probability and give $d^\top \mathbf{1}_{\{i\}} = d_i$. These values cannot all be equal: together with $\sum_i d_i = 0$, equality would imply $d = 0$. Hence $d^\top \mathbf{1}_S$ has positive variance. Thus $d^\top \mathrm{Cov}(\mathbf{1}_S) d > 0$, so the Hessian is positive definite on the tangent space $\{d \in \mathbb{R}^n : \mathbf{1}^\top d = 0\}$. □

C.2 Projected-gradient convergence rates

At fixed temperature, the smoothness bound gives the following objective-error rates for ordinary and accelerated projected gradient descent. Both methods preserve efficiency.

Proposition C.2 (Projected-gradient ELC)**.** Let x_T be the output after $T \ge 1$ iterations of Equation (13) initialized at x_0 with $x_0(N) = v(N)$. Then the following hold:

1. The update preserves efficiency; hence, $x_T(N) = v(N)$.
2. With step size $\alpha = 1/\beta = 4\eta/n$, if $x^\star$ is the unique minimizer of Equation (9), then
$$\varepsilon_\eta(x_T) - \varepsilon_\eta(x^\star) \le \frac{n}{8\eta T}\,\|x_0 - x^\star\|_2^2.$$
3. Applying Nesterov acceleration to the same projected-gradient problem also preserves efficiency and improves the standard smooth-convex rate to
$$\varepsilon_\eta(x_T) - \varepsilon_\eta(x^\star) \le \frac{n}{2\eta(T+1)^2}\,\|x_0 - x^\star\|_2^2.$$

Proof of Proposition C.2. The update direction in Equation (13) has zero sum because

$$\sum_{i=1}^{n}\big(r_i(x) - \bar{r}(x)\big) = 0. \tag{19}$$

Thus the update preserves $x(N)$, and an efficient initialization remains efficient after any number of iterations.

By Lemma C.1, ε_η is β-smooth on the efficiency hyperplane with $\beta = n/(4\eta)$. Projected gradient descent with step size $\alpha = 1/\beta$ over an affine subspace therefore satisfies the standard smooth convex rate (Nesterov, 2018)

$$\varepsilon_\eta(x_T) - \varepsilon_\eta(x^\star) \le \frac{\beta}{2T}\|x_0 - x^\star\|_2^2 = \frac{n}{8\eta T}\|x_0 - x^\star\|_2^2. \tag{20}$$

Nesterov acceleration over the same affine subspace preserves efficiency because it only takes affine combinations of efficient payoff vectors and projected gradient steps with zero-sum directions. The standard accelerated smooth convex guarantee gives (Nesterov, 2018)

$$\varepsilon_\eta(x_T) - \varepsilon_\eta(x^\star) \le \frac{2\beta}{(T+1)^2}\|x_0 - x^\star\|_2^2 = \frac{n}{2\eta(T+1)^2}\|x_0 - x^\star\|_2^2. \tag{21}$$

□

Corollary C.3 (Maximum-excess accuracy)**.** Let $n \geq 2$, $\mathcal{S} = 2^N \setminus \{\emptyset, N\}$, and $M = |\mathcal{S}|$. At fixed temperature $\eta > 0$, let x_T be the accelerated iterate of Proposition C.2 from an efficient initial payoff x_0, and set $R_\eta = \|x_0 - x_\eta\|_2$. For $T \geq 1$,

$$0 \leq \hat{\varepsilon}(x_T) - \varepsilon^* \leq \eta \log M + \frac{nR_\eta^2}{2\eta(T+1)^2}.$$

In particular, for any $\delta > 0$, the choices

$$\eta = \frac{\delta}{2\log M}, \qquad T + 1 \geq \frac{R_\eta \sqrt{2n \log M}}{\delta}$$

ensure $\hat{\varepsilon}(x_T) - \varepsilon^* \leq \delta$.

Proof. Let x^{LC} be a least-core payoff, which exists by Lemma B.2. The log-sum-exp bounds and optimality of x_η give

$$\hat{\varepsilon}(x_T) \leq \varepsilon_\eta(x_T), \qquad \varepsilon_\eta(x_\eta) \leq \varepsilon_\eta(x^{\mathrm{LC}}) \leq \varepsilon^* + \eta \log M.$$

Combining these inequalities with the accelerated rate in Proposition C.2 proves the upper bound. Efficiency of x_T gives the lower bound. The stated choices make each upper-bound term at most $\delta/2$. □

Since $M = 2^n - 2$, this gives a sufficient iteration budget of $O(1 + nR_\eta/\delta)$. The distance factor R_η may itself depend on dimension and temperature; the bound is not a necessary iteration count. It controls the maximum-excess objective error, not payoff distance to the nucleolus, and applies at a fixed temperature rather than bounding the total cost of the annealing schedule.

D Experimental protocol details

This section supplements the quality, selector, and player-identification experiments in Sections 5 and 6 with solver settings and a check of sensitivity to initialization.

ELC starts at $x_0 = (v(N)/n)\mathbf{1}$ and uses Nesterov acceleration with step size $4\eta/n$. The nine temperatures are $\eta \in \{1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.002\}$. Each stage starts from the preceding payoff, with momentum reset, and stops at a projected-gradient infinity norm of 10^{-8} or 10,000 iterations. An iteration-limited payoff initializes the next stage.

Numerical stability. We evaluate the objective and Gibbs weights using stabilized log-sum-exp and softmax routines. For $z_S = e_S(x)/\eta$ and $m = \max_S z_S$, these compute

$$\log \sum_S e^{z_S} = m + \log \sum_S e^{z_S - m}, \qquad p_S = \frac{e^{z_S - m}}{\sum_T e^{z_T - m}}.$$

The exponentials therefore have nonpositive arguments, avoiding overflow from directly exponentiating large scaled excesses. The implementation subtracts the constant $\eta \log M$ from the objective in Equation (7); this leaves its gradient and minimizer unchanged.

The initial least-core LP in runtime comparisons uses HiGHS interior point with crossover and primal and dual feasibility tolerances of 10^{-9}. Minimum-norm selection uses SLSQP, with constraint generation for larger coalition families; its numerical relaxation is specified in Section F.2.

Timing repetitions randomize method order within each game. Warm-up computations are excluded, CUDA is synchronized at timing boundaries, and numerical-library threads are limited to one per worker. Standard errors use the sample standard deviation divided by the square root of the number of independent games or splits.

Sensitivity to initialization. For each of 60 random games, we run the same annealing path from five initializations: the uniform payoff, LP + min-norm, a TMC Shapley estimate, and two random efficient payoffs. Endpoints agree to about 10^{-12} within each game (Table 6), indicating negligible initialization sensitivity under this protocol. The table also records how often the iteration limit is reached.

Table 6: Endpoint variability across five initializations under nine-stage annealing. Diameter is the maximum pairwise endpoint ℓ_2 distance; hard gap is maximized over the five starts. Values use 20 random games per size. The last two columns count games reaching an iteration limit at any stage or at the final stage, respectively.

n	Games	Diameter ($\times 10^{-12}$)		Max hard gap ($\times 10^{-4}$)	Capped games	
		Mean $\pm$ SE	Max	Mean $\pm$ SE	Any stage	Final stage
6	20	1.269 ± 0.350	5.594	4.605 ± 0.998	4	0
7	20	1.007 ± 0.382	5.776	7.069 ± 1.059	4	2
8	20	2.275 ± 0.451	5.439	4.836 ± 1.166	11	4

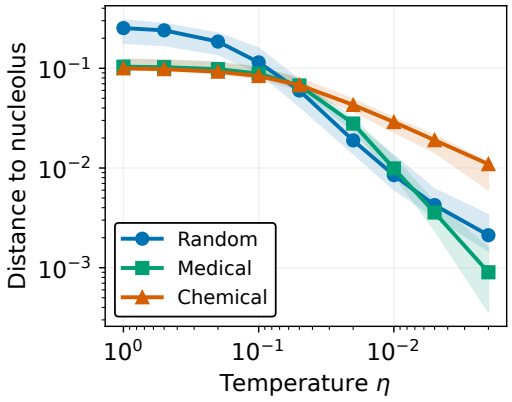

Figure 1: Annealing toward the nucleolus: median and interquartile range for 60 random games, ten medical splits, and nine chemical splits. Iteration-limited paths are retained; one chemical split lacks a valid reference and is excluded. Temperature decreases left to right.

E Annealing Trajectories

We examine how the computed payoff approaches the nucleolus during annealing in the random and UCI games of Section 5. Median distance decreases at every measured temperature transition in all three settings (Figure 1). Individual paths follow the same trend except in four of 60 random games, where the largest upward step is 0.00279. All displayed medical and chemical paths decrease monotonically up to a 10^{-10} numerical tolerance.

F Computational Accuracy and Scaling

We detail the accuracy and timing comparisons in Section 5, then examine GPU acceleration and coalition sampling at larger sizes.

F.1 Accuracy and Runtime Relative to the Nucleolus

The comparison in Table 3 measures how cheaply ELC approximates the nucleolus. It uses 20 random games per size $n = 6, 7, 8$ and ten splits each of medical, chemical, and house, with three timing repetitions per game. Distances and gaps are medians of within-game repetition medians; times average all finished repetitions.

ELC time covers all nine annealing stages; nucleolus time covers the initial least-core LP and subsequent lexicographic optimization. Both exclude shared game construction and independent accuracy evaluation. This compares attained accuracy and runtime, rather than time to a common distance target. The largest absolute least-core gap of the nucleolus references is 3.78×10^{-15}.

House distances use saved references for all ten splits, but its nucleolus runtime uses only 15 completed repetitions; 15 worker timeouts under a 3600s cap are excluded. One chemical split is excluded from min-norm distances because its reference solve failed. ELC trajectories reaching a stage iteration limit remain included.

F.2 CPU/GPU SCALING AND SAMPLED BASELINES

Setup. Table 7 uses the hardware and solvers in Sections D and 5, with identical float64 updates for CPU/GPU ELC and a 600s solver limit. For $n \in \{8, 10, \ldots, 24\}$, each game has $n-2$ useful players, two distractors, $\tau = n - 2$, $a_j = 0.1(n-2)$, and $b = h_j = 2(n-2)$. Useful effects are uniform in $[0.8, 1.2]$, rescaled to sum to $n-2$. We use five games and three timing repetitions per condition. The 10%/50% families are nested uniform samples with at least $4n$ rows, shared across methods and repetitions. For each constraint family $\mathcal{A}$, min-norm selection uses $e_S(x) \le t_{\mathcal{A}} + 10^{-9} q_{\mathcal{A}}$, where $q_{\mathcal{A}} = \max(1, |v(N)|, \max_{S \in \mathcal{A}} |v(S)|)$ and $t_{\mathcal{A}}$ is the maximum excess of the verified initial LP allocation.

Reading the table. Time averages all 15 runs: measured solver times for successes and numerical failures, and 600s for timeouts (+ denotes a lower bound). It includes the initial LP and internal verification for min-norm, but excludes data preparation, oracle evaluation, transfers, warm-up, and independent final evaluation. Num. err. counts numerical solver failures or rejected numerical solutions. Gap and AP average successful runs only; a dash means none succeeded. Gap is $\max(0, \hat{\varepsilon}(x) - \epsilon^{\star}_{\text{full}})/q_{\mathcal{S}}$, evaluated on the full family; AP follows Section 6. Successful subsets may differ across methods: both sampled min-norm conditions at $n = 20$ and the 10% condition at $n = 22$ represent only one successful game, repeated three times.

Results. GPU ELC is $1.55\times$ and $3.14\times$ faster than CPU ELC at $n = 20, 22$, respectively, where both complete all runs. At $n = 24$, GPU ELC finishes all runs while CPU ELC always times out. These comparisons use attained accuracies, not a common accuracy target. Sampled LP is faster than full LP but has larger full-family gaps; adding min-norm selection often loses the runtime advantage. For $n \in \{20, 22, 24\}$ across both sampling budgets, the initial LP succeeds in all 90 sampled min-norm runs. All 27 runs containing the clean coalition U (all useful players and no distractors) succeed; without U, 60 runs time out during SLSQP-based selection and three report SLSQP numerical failures. These are nine and 21 sampled game settings, respectively, each repeated three times. This association localizes the failures to the selection stage but does not establish the numerical cause or an intrinsic disadvantage of the min-norm selector.

Table 7: Computational scaling and attribution on contextual-distractor games. Time averages all 15 runs; Gap and AP average successful runs only. + marks a time lower bound; – in Gap or AP indicates no successful runs. Settings and aggregation are described in Section F.2.

n	Method	Time (s)	Gap	AP$^+$	AP$^-$	Timeout/15	Num. err./15
8	LP	0.003	0	1.000	1.000	0/15	0/15
8	LP + min-norm	0.005	1e-09	1.000	1.000	0/15	0/15
8	50% LP	0.003	0.2	1.000	1.000	0/15	0/15
8	50% LP + min-norm	0.004	0.2	1.000	1.000	0/15	0/15
8	10% LP	0.003	0.32	1.000	1.000	0/15	0/15
8	10% LP + min-norm	0.004	0.278	1.000	1.000	0/15	6/15
8	ELC CPU	0.009	2.29e-14	1.000	1.000	0/15	0/15
8	ELC GPU	0.029	2.12e-14	1.000	1.000	0/15	0/15
10	LP	0.007	0	1.000	1.000	0/15	0/15
10	LP + min-norm	0.008	1e-09	1.000	1.000	0/15	0/15
10	50% LP	0.006	0.0792	0.997	0.967	0/15	0/15
10	50% LP + min-norm	0.007	0.0792	1.000	1.000	0/15	0/15
10	10% LP	0.003	0.28	0.995	0.917	0/15	0/15
10	10% LP + min-norm	0.004	0.28	0.995	0.917	0/15	0/15
10	ELC CPU	0.013	2.61e-13	1.000	1.000	0/15	0/15
10	ELC GPU	0.037	2.82e-13	1.000	1.000	0/15	0/15

Continued on next page

Table 7 (continued)

n	Method	Time (s)	Gap	AP^+	AP^-	Timeout/15	Num. err./15
12	LP	0.024	0	0.957	0.669	0/15	0/15
12	LP + min-norm	0.026	1e-09	1.000	1.000	0/15	0/15
12	50% LP	0.017	0.267	0.909	0.316	0/15	0/15
12	50% LP + min-norm	0.023	0.267	0.918	0.383	0/15	0/15
12	10% LP	0.006	0.276	0.918	0.396	0/15	0/15
12	10% LP + min-norm	0.008	0.276	0.918	0.383	0/15	0/15
12	ELC CPU	0.030	4.95e-13	1.000	1.000	0/15	0/15
12	ELC GPU	0.059	4.4e-13	1.000	1.000	0/15	0/15
14	LP	0.116	9.87e-18	0.866	0.578	0/15	0/15
14	LP + min-norm	0.122	1e-09	1.000	1.000	0/15	0/15
14	50% LP	0.073	0.204	0.808	0.390	0/15	0/15
14	50% LP + min-norm	0.234	0.204	0.836	0.474	0/15	0/15
14	10% LP	0.015	0.333	0.719	0.112	0/15	0/15
14	10% LP + min-norm	0.023	0.333	0.719	0.112	0/15	0/15
14	ELC CPU	0.088	6.46e-15	1.000	1.000	0/15	0/15
14	ELC GPU	0.164	0	1.000	1.000	0/15	0/15
16	LP	0.527	2.54e-17	0.870	0.567	0/15	0/15
16	LP + min-norm	0.548	1e-09	1.000	1.000	0/15	0/15
16	50% LP	0.388	0.07	0.868	0.559	0/15	0/15
16	50% LP + min-norm	12.181	0.07	0.946	0.819	0/15	0/15
16	10% LP	0.059	0.333	0.731	0.096	0/15	0/15
16	10% LP + min-norm	0.474	0.333	0.731	0.096	0/15	0/15
16	ELC CPU	0.408	1.45e-15	1.000	1.000	0/15	0/15
16	ELC GPU	0.633	2.96e-15	1.000	1.000	0/15	0/15
18	LP	2.405	0	0.879	0.559	0/15	0/15
18	LP + min-norm	2.516	1e-09	1.000	1.000	0/15	0/15
18	50% LP	1.654	0.136	0.852	0.458	0/15	0/15
18	50% LP + min-norm	127.094^+	0.0868	0.938	0.771	3/15	0/15
18	10% LP	0.267	0.333	0.751	0.095	0/15	0/15
18	10% LP + min-norm	29.617	0.333	0.751	0.095	0/15	0/15
18	ELC CPU	2.632	1.67e-14	1.000	1.000	0/15	0/15
18	ELC GPU	2.470	0	1.000	1.000	0/15	0/15
20	LP	12.645	0	0.886	0.553	0/15	0/15
20	LP + min-norm	13.277	1e-09	1.000	1.000	0/15	0/15
20	50% LP	7.134	0.267	0.791	0.176	0/15	0/15
20	50% LP + min-norm	481.904^+	1e-09	1.000	1.000	12/15	0/15
20	10% LP	1.306	0.271	0.814	0.261	0/15	0/15
20	10% LP + min-norm	432.288^+	0.0196	1.000	1.000	9/15	3/15
20	ELC CPU	15.129	2.36e-14	1.000	1.000	0/15	0/15
20	ELC GPU	9.769	4.8e-14	1.000	1.000	0/15	0/15
22	LP	62.547	0	0.893	0.548	0/15	0/15
22	LP + min-norm	65.457	1e-09	1.000	1.000	0/15	0/15
22	50% LP	44.586	0.138	0.891	0.537	0/15	0/15
22	50% LP + min-norm	272.163^+	0.00723	1.000	1.000	6/15	0/15
22	10% LP	7.322	0.27	0.824	0.222	0/15	0/15
22	10% LP + min-norm	482.577^+	0.0167	1.000	1.000	12/15	0/15
22	ELC CPU	145.965	1.58e-14	1.000	1.000	0/15	0/15
22	ELC GPU	46.525	4.98e-14	1.000	1.000	0/15	0/15
24	LP	317.264	0	0.898	0.543	0/15	0/15
24	LP + min-norm	330.111	1e-09	1.000	1.000	0/15	0/15
24	50% LP	180.193	0.133	0.856	0.355	0/15	0/15
24	50% LP + min-norm	364.165^+	1e-09	1.000	1.000	6/15	0/15
24	10% LP	25.628	0.333	0.793	0.063	0/15	0/15
24	10% LP + min-norm	600.000^+	–	–	–	15/15	0/15
24	ELC CPU	600.000^+	–	–	–	15/15	0/15
24	ELC GPU	214.134	0	1.000	1.000	0/15	0/15

G GENERALIZED CONTEXTUAL-DISTRACTOR EXPERIMENTS

We supplement the player-identification results in Section 6 with game generation and evaluation details. Additional tables and figures separate equal-harm games and show how identification changes with the context threshold.

G.1 GAMES AND SAMPLING

The games in Table 4 follow Equation (14) with $(n, |U|, |D|) = (10, 6, 2)$ and $(14, 8, 3)$, leaving two and three neutral players, respectively. Useful effects are drawn independently from $[0.8, 1.2]$ and rescaled to sum to $|U|$.

The main-text experiments use heterogeneous harm: equally spaced h_j/b from 0.25 to 0.75, permuted across distractors. We additionally test the homogeneous special case in which every $h_j = b$. In both cases $a_j = 0.1|U|$ and $b = 2|U|$. The thresholds are $3, 5, 6$ for $n = 10$ and $4, 6, 8$ for $n = 14$. Ten game seeds per size and threshold give 60 games for the main setting and 60 for the homogeneous special case.

Each game uses all $M = 2^n - 2$ proper coalitions or a uniform sample without replacement of $\lfloor \rho M \rfloor$ rows for $\rho \in \{0.1, 0.5\}$. Five sample seeds per game give 1,200 sampled conditions. The two budgets are nested prefixes of the same random permutation. No targeted rows or minimum-row adjustment are used. LP, LP + min-norm, ELC, and the Shapley estimator receive identical sampled rows and values. ELC uses the shared annealing protocol in Section D.

The equal-count variant in Table 9 instead uses $(n, |U|, |D|) = (10, 4, 4)$ and $(14, 6, 6)$, with two neutrals and thresholds $\{2, 3, 4\}$ and $\{3, 5, 6\}$, respectively. It uses the same two harm settings, sample construction, and solvers. This changes both group sizes and the contexts in which harm occurs, so differences between these grids do not isolate a causal effect of group size.

For each grid, all methods return 120 full-game results. Of 1,200 sampled conditions, 32 LP + min-norm solves fail numerical acceptance and are excluded; the other methods return all 1,200 outputs. In the equal-count grid, 31 failures occur at 10% and one at 50%. ELC returns finite iterates even when a stage reaches its iteration limit; in the original grid, all full endpoints and 1,199 sampled endpoints meet the final-stage gradient tolerance.

G.2 EVALUATION METRICS AND AGGREGATION

We measure group identification, negative-credit assignment, and within-group ordering separately. AP^+ and AP^- rank the useful and distractor groups by x_i and $-x_i$, using grouped tied scores. Negative credit means $x_i < -10^{-7}$. For both means and SE, sample seeds are averaged within each game first; games are the independent units. Failed solves are excluded, with availability specified in Section G.1.

Within-group strength order. For useful pairs, we compare $m_i - m_k$ with $x_i - x_k$; for distractor pairs, we compare $(h_j - a_j) - (h_\ell - a_\ell)$ with $-x_j + x_\ell$. Only pairs with distinct generating strengths enter the correct, tied, and reversed fractions; order accuracy is the correct fraction. The tie tolerance is $10^{-7} \max(1, \|s\|_\infty)$ for a group's strength vector s, applied to both target and payoff differences. Equal-harm distractors have no target order and enter only identification and sign metrics.

For the full-coalition game in Table 1, symmetry and additive translation explain the common $+7.5$ useful-player offset: subtracting the additive useful contributions leaves a game symmetric in useful players, so ELC assigns them a common interaction share. Subtracting this share would violate efficiency. Asymmetric interactions need not produce a common offset.

G.3 SHAPLEY ESTIMATION ON MATCHED COALITION SAMPLES

We compute sampled Shapley from the identical uniform-without-replacement coalition rows used by LP, LP + min-norm, and ELC. This controls the available coalition information across methods. The comparison concerns this matched-row estimator rather than alternative Shapley sampling schemes.

Table 8: Equal-harm counterpart to Table 4: $h_j = b$ for all distractors, with 60 games and the same metrics and aggregation. Higher is better; bold marks the best mean within each budget, including ties. Harm order is undefined and shown as a dash.

Constraints	Method	AP^+	AP^-	Distractor negative rate	Useful order accuracy	Harm order accuracy
Full	Shapley	**1.000**	0.750	0.500	**1.000**	—
	LP	**1.000**	0.698	0.417	0.909	—
	LP + min-norm	**1.000**	**1.000**	**1.000**	0.667	—
	ELC	**1.000**	**1.000**	**1.000**	**1.000**	—
50% sample	Shapley	0.884	0.562	0.506	0.566	—
	LP	0.995	0.719	0.482	**0.931**	—
	LP + min-norm	0.996	**0.979**	**0.993**	0.676	—
	ELC	**1.000**	0.969	**0.993**	0.903	—
10% sample	Shapley	0.861	0.489	0.389	0.549	—
	LP	0.980	0.752	0.625	**0.807**	—
	LP + min-norm	**0.985**	**0.842**	**0.794**	0.775	—
	ELC	0.984	0.832	0.786	0.772	—

Let $\mathcal{B} \subset \mathcal{S}$ contain m of the $M = 2^n - 2$ proper coalitions. For $k = |S|$, define

$$c_i(S) = \begin{cases} 1/[k\binom{n}{k}], & i \in S, \\ -1/[(n-k)\binom{n}{k}], & i \notin S. \end{cases}$$

Collecting the coefficients of each coalition value in the marginal contribution definition gives the full Shapley formula. Since each row has inclusion probability m/M, its inverse-probability estimator is

$$\widehat{\phi}_i = \frac{v(N) - v(\emptyset)}{n} + \frac{M}{m} \sum_{S \in \mathcal{B}} c_i(S)\,[v(S) - v(\emptyset)].$$

It is unbiased under the specified sampling design. Also, $\sum_i c_i(S) = 0$ for every proper row, so every output sums to $v(N) - v(\emptyset)$; here $v(\emptyset) = 0$. Full enumeration recovers the exact value. Inputs are only the sampled coalition rows and values and the two endpoint values.

G.4 Identification across game settings

We test how the identification patterns change with group sizes, equal distractor harm, and the context threshold. Tables 8 and 10 set $h_j = b$ for every distractor in the original and equal-count grids. Full ELC and LP + min-norm identify both groups perfectly in both grids, but sampling lowers AP^- while retaining high AP^+. ELC also preserves useful-player ordering under the full family, whereas min-norm selection partly equalizes their payoffs.

Equal useful and distractor counts. Table 9 checks whether the sampling effect persists when the two groups are equally numerous. AP^- still falls for Shapley, ELC, and LP + min-norm; for the latter two, AP^+ remains near one. The game parameters and the limits of comparison with the original grid are given in Section G.1.

The threshold comparisons in Figures 2 and 3 show that distractor identification generally worsens as harmful contexts become rarer, while useful players remain better identified. Larger samples usually improve recovery but still fall short of full ELC and LP + min-norm. Full Shapley also loses distractor identification at higher thresholds despite preserving useful-player rankings.

H Downstream attribution checks

These experiments assess whether attribution rankings help with training-point removal, feature removal, and mislabeled-point detection. We expand the training-data valuation results in Section 7 and then examine the two additional tasks.

Table 9: Equal useful and distractor counts with varying distractor harm. As in Table 4, h_j/b ranges from 0.25 to 0.75 across distractors, but group sizes are now equal: $(n, |U|, |D|) = (10, 4, 4)$ and $(14, 6, 6)$, with two neutrals. Means use 60 games; metrics, aggregation, and boldface follow Table 4. Game generation and exclusion rules are in Section G.1.

Constraints	Method	AP^+	AP^-	Distractor negative rate	Useful order accuracy	Harm order accuracy
Full	Shapley	**1.000**	0.673	0.208	**1.000**	**1.000**
	LP	**1.000**	0.761	0.256	0.897	0.373
	LP + min-norm	**1.000**	**1.000**	**1.000**	0.667	0.000
	ELC	**1.000**	**1.000**	**1.000**	**1.000**	0.633
50% sample	Shapley	0.845	0.654	0.429	0.572	0.593
	LP	0.996	0.823	0.492	**0.860**	0.334
	LP + min-norm	0.995	**0.974**	**0.977**	0.667	0.092
	ELC	**1.000**	0.953	0.940	0.845	**0.620**
10% sample	Shapley	0.787	0.609	0.267	0.564	0.550
	LP	0.959	0.861	0.712	**0.623**	0.515
	LP + min-norm	0.961	**0.891**	**0.814**	0.571	0.473
	ELC	**0.968**	0.878	0.787	0.605	**0.616**

Table 10: Equal useful and distractor counts with identical distractor harm. Group sizes match Table 9, but every distractor has $h_j = b$, so any one distractor removes the entire bonus. Means use 60 games with the same metrics, aggregation, and boldface. Harm order is undefined because all distractors are equally harmful and is shown as a dash.

Constraints	Method	AP^+	AP^-	Distractor negative rate	Useful order accuracy	Harm order accuracy
Full	Shapley	**1.000**	0.764	0.167	**1.000**	—
	LP	**1.000**	0.757	0.208	0.896	—
	LP + min-norm	**1.000**	**1.000**	**1.000**	0.667	—
	ELC	**1.000**	**1.000**	**1.000**	**1.000**	—
50% sample	Shapley	0.804	0.603	0.370	0.574	—
	LP	0.995	0.794	0.418	**0.870**	—
	LP + min-norm	0.991	**0.976**	**0.987**	0.676	—
	ELC	**1.000**	0.952	0.948	0.844	—
10% sample	Shapley	0.746	0.569	0.198	0.560	—
	LP	0.958	0.783	0.559	**0.726**	—
	LP + min-norm	0.961	**0.821**	**0.687**	0.657	—
	ELC	**0.965**	0.808	0.646	0.696	—

H.1 Training-point removal curves

The Gaussian and Dogfish curves show how prediction accuracy changes when training points are removed in attribution order (Figures 4 and 5). ELC's mean normalized AUC improves on both baselines across all budgets and removal directions, except against min-norm on Dogfish best-first removal at 10K queries. The Gaussian gains are larger; several of the smaller gains have confidence intervals that include zero.

For the paired AUC differences in Table 5, we bootstrap repetition-level differences 10,000 times to obtain 95% confidence intervals. The curve bands instead show sample standard deviations of accuracy, using 20 Gaussian and five Dogfish repetitions. The two displays therefore summarize different kinds of uncertainty.

H.2 Feature removal on UCI datasets

This experiment tests feature rankings by removing input features and refitting a logistic-regression classifier. Coalition values are held-out accuracies on ten fixed 80/20 stratified splits, with matched

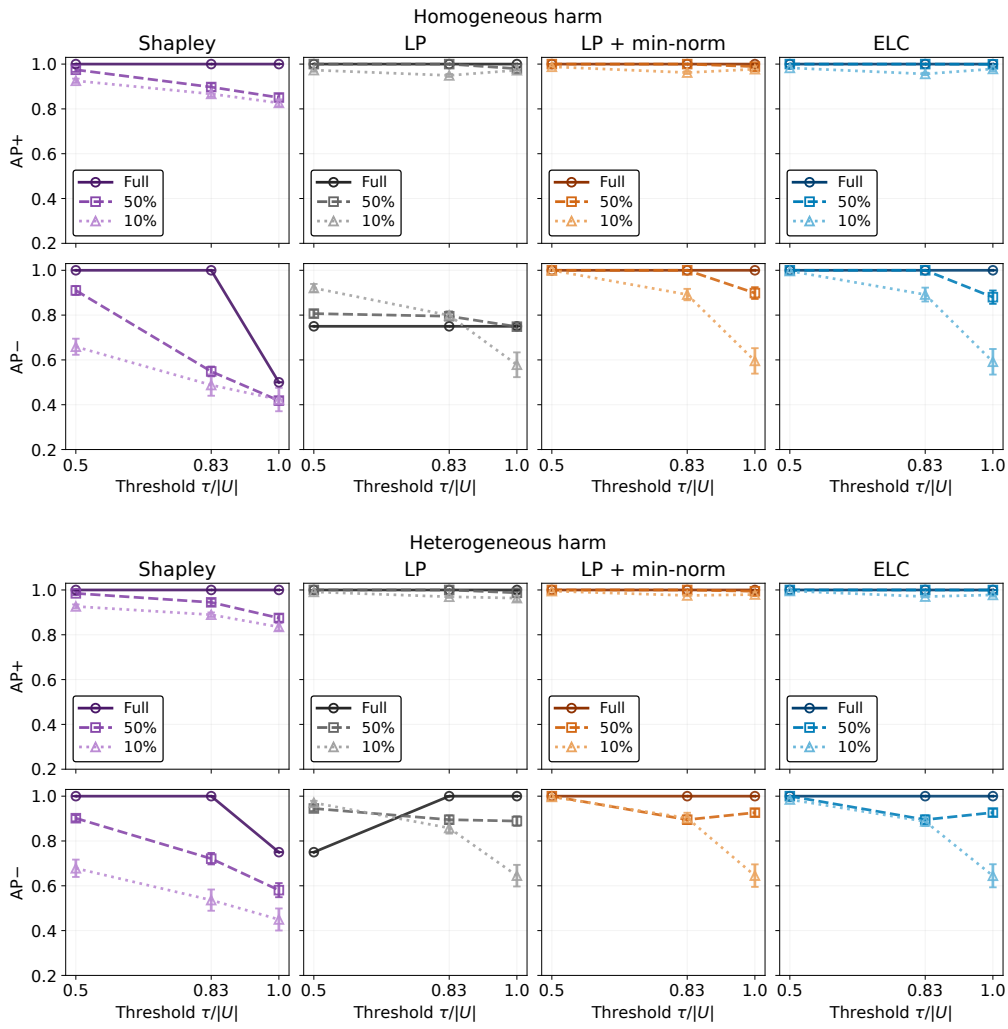

Figure 2: Threshold comparisons at $n = 10$, with homogeneous harm above and heterogeneous harm below. Columns compare methods; each pair of rows shows AP^+ and AP^-. Full, 50%, and 10% coalition families use dark-to-light shades and distinct line styles and markers. Sampled Shapley uses the estimator in Section G.3. Means and SEs use ten games after averaging samples within each game.

preprocessing across methods. Removal ranges from zero to $n - 1$ features; Table 11 averages the ELC-minus-min-norm accuracy difference over these steps and then across splits.

Mean accuracy differences favor ELC in both removal directions on Chemical. Medical shows almost no mean difference, while the means on House favor ELC for lowest-first removal and min-norm for highest-first removal. These tasks evaluate predictive utility rather than recovery of labeled contextual distractors.

H.3 Label correction on Enron

We adopt the Enron label-correction task of Yan & Procaccia (2021) to test whether low attribution scores identify mislabeled training examples. We use 1,000 Enron emails with character n-gram count features and Multinomial Naive Bayes. The 489 training points are players; 20% of training labels are randomly flipped. Each of five repetitions uses 5,000 coalition queries, with Shapley estimated by truncated Monte Carlo.

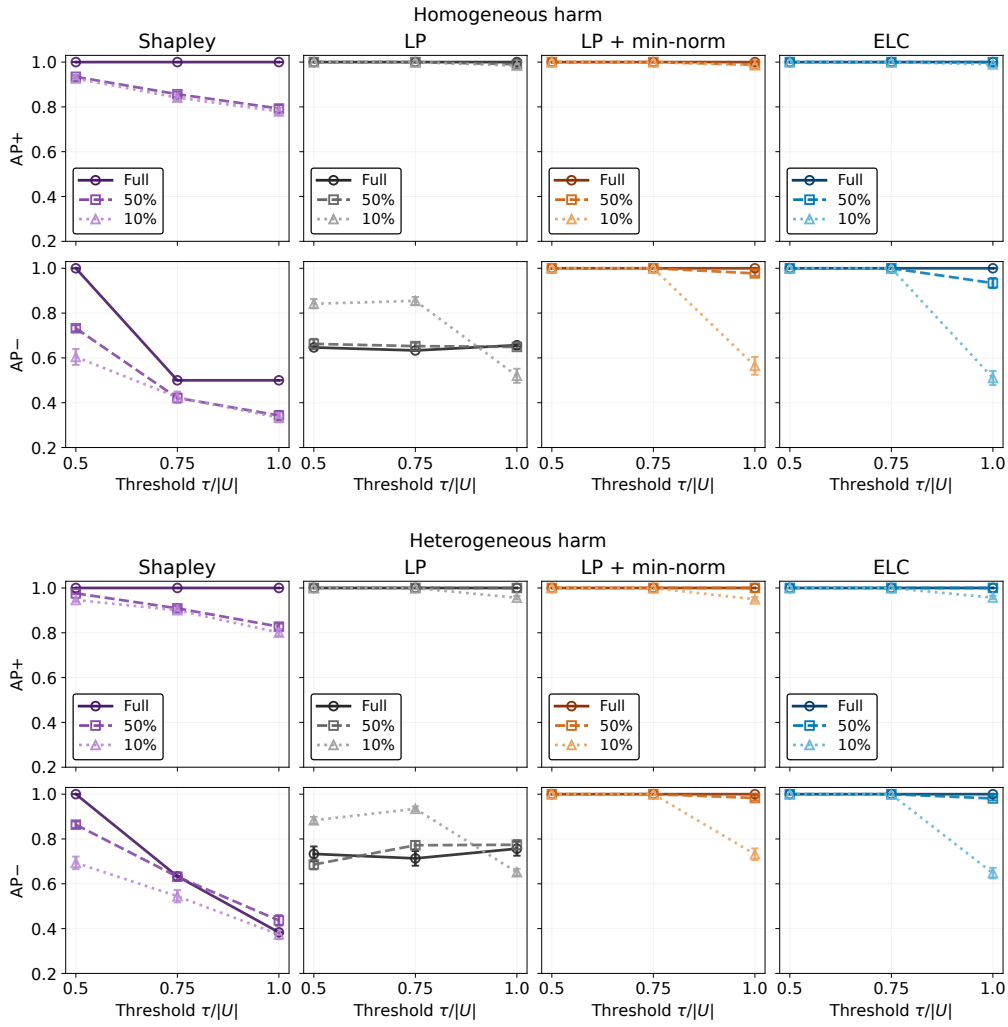

Figure 3: Threshold comparisons at $n = 14$, with the same layout, aggregation, and line conventions as Figure 2.

Table 11: Feature-removal accuracy differences, ELC minus (LP + min-norm), in percentage points: each split contributes its mean over removal steps, followed by mean ± SE over ten splits. Positive favors ELC for lowest-first removal; negative favors ELC for highest-first removal.

Dataset	Lowest first	Highest first
Medical	-0.032 ± 0.056	$+0.000 \pm 0.068$
Chemical	$+0.299 \pm 0.214$	-0.449 ± 0.217
House	$+0.029 \pm 0.031$	$+0.244 \pm 0.242$

We restore labels in increasing payoff order. Correction AUC integrates accuracy over 0–35% restored points and divides by 0.35. Mislabel AP ranks $-x_i$ against the flipped-label indicator. Flip recall is measured among the lowest-ranked points, selecting as many points as there are flipped labels.

ELC has the highest mean correction AUC, with a small margin over LP + min-norm and Shapley (Table 12). Shapley is strongest at identifying flipped labels, while ELC and min-norm have similar correction curves (Figure 6). It is worth noting that mislabeled samples are not necessarily contextual distractors. Contextual distractors are those becoming harmful at some context and otherwise

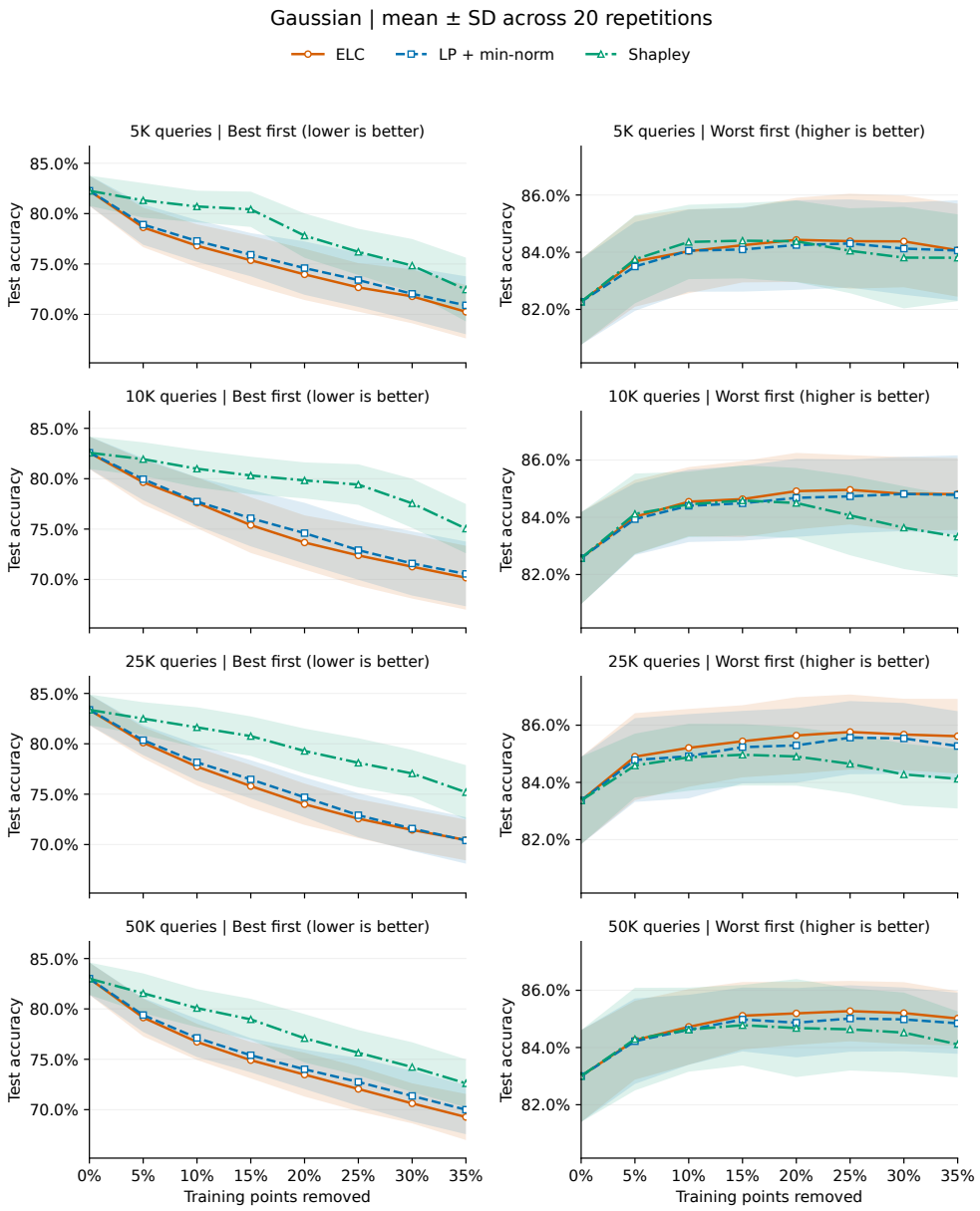

Figure 4: Gaussian training-point removal at all four query budgets. Lines are means over 20 repetitions and bands are ± 1 sample standard deviation. Best-first removal favors lower accuracy; worst-first removal favors higher accuracy. Vertical scales are shared across budgets within each removal direction.

harmless, while mislabeled samples generally do not have such inconsistency. In Yan & Procaccia (2021), the performance of the Shapley value was not reported.

I Reference experiment: image-marker attribution

The image-marker task of Nagarajan et al. (2026) tests attribution to a visually identifiable contextual distractor. A ResNet-50 classifier predicts cat, dog, or bird; a red marker is added to all dog images and 30% of cat images during dataset construction. We evaluate 100 correctly classified cats

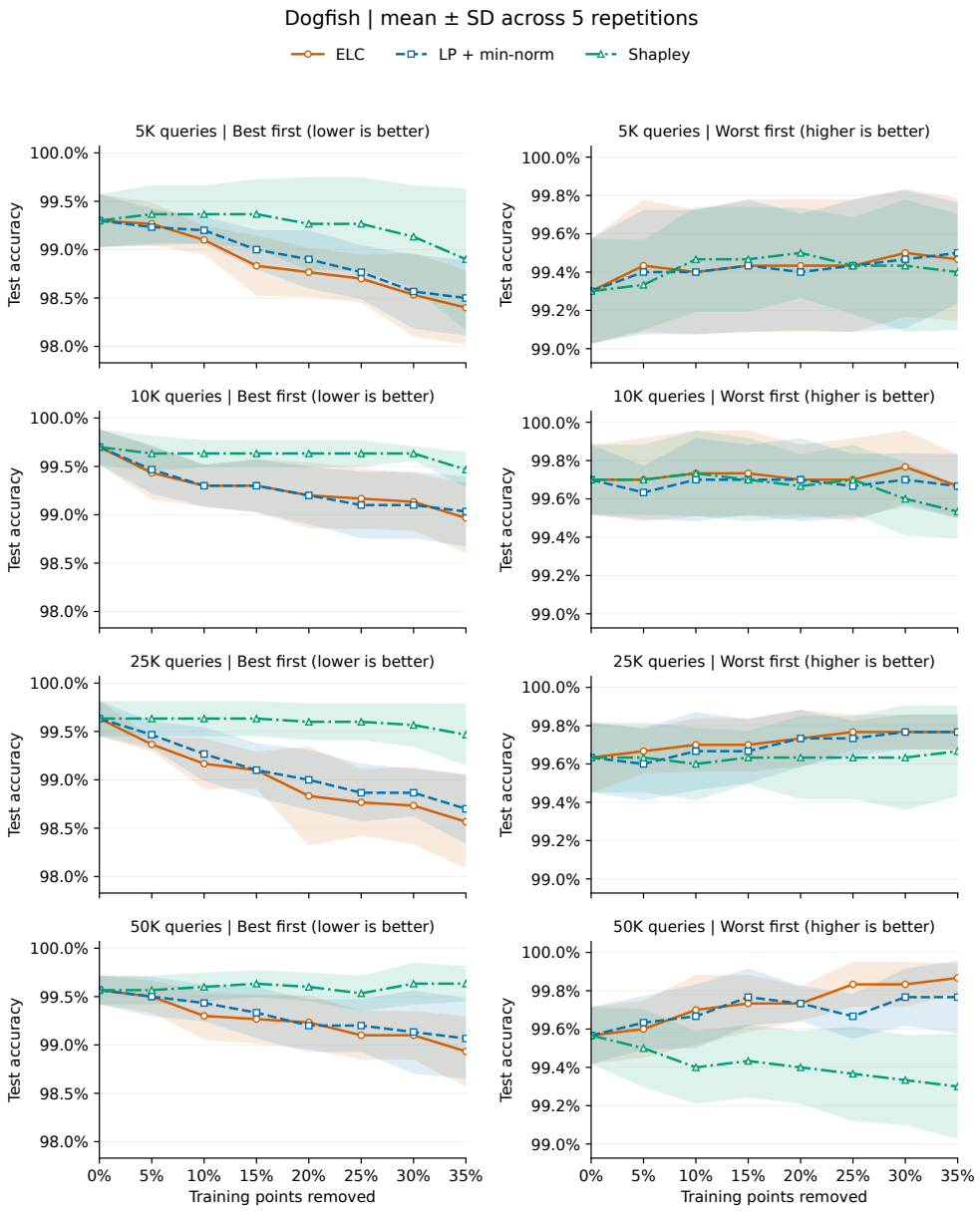

Figure 5: Dogfish training-point removal at all four query budgets. Lines are means over five repetitions and bands are ± 1 sample standard deviation. Layout and scale conventions match Figure 4.

containing the marker, using one sampling seed. Images are not selected by the sign of the marker's contribution.

Each 224×224 image is divided into an 8×8 grid of player patches. The group D comprises patches overlapping the marker. Coalition values are target-class logits, with absent patches replaced by black and the empty-image logit subtracted. Removing D therefore removes entire patches, including pixels outside the marker.

Shapley uses 100 permutations. LP, LP + min-norm, and ELC share 6,400 sampled coalitions with independent patch-inclusion probability 0.5, plus the empty and grand coalitions. LP + min-norm uses CVXPY to solve an LP followed by a quadratic program. ELC uses nine annealing stages with

Table 12: Enron label-correction and mislabel-identification results (mean ± SE over five repetitions). Bold and underline mark the highest and second-highest mean in each performance column, respectively.

Method	Corr. norm. AUC	Corr. acc. at 35%	Mislabel AP	Flip recall
Random	0.7200 ± 0.011	0.723 ± 0.013	0.211 ± 0.005	0.204 ± 0.018
Shapley	0.7306 ± 0.011	$\mathbf{0.743} \pm 0.012$	$\mathbf{0.456} \pm 0.023$	$\mathbf{0.433} \pm 0.036$
LP + min-norm	$\underline{0.7310} \pm 0.013$	0.741 ± 0.012	0.268 ± 0.015	0.273 ± 0.016
ELC	$\mathbf{0.7336} \pm 0.013$	$\underline{0.742} \pm 0.012$	$\underline{0.274} \pm 0.015$	$\underline{0.286} \pm 0.014$

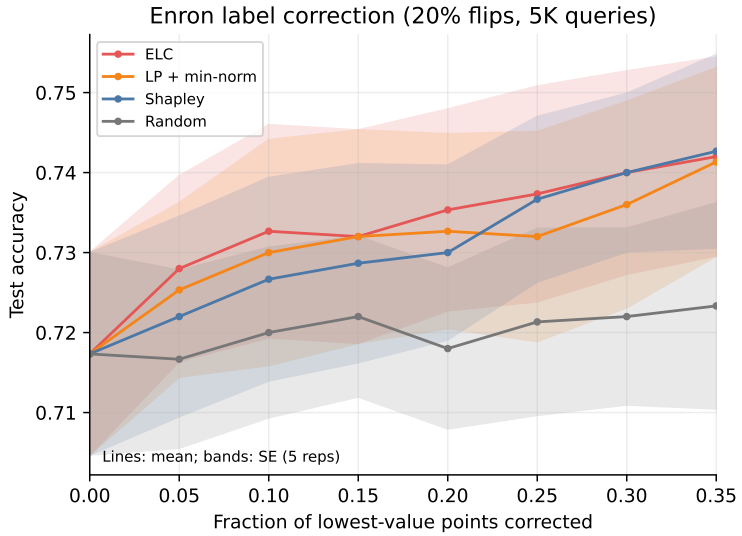

Figure 6: Enron label correction after restoring an increasing fraction of the lowest-valued training points. Curves are means over five repetitions and bands are mean ± SE.

at most 10,000 updates each. All 100 ELC runs reach the iteration limit, with final projected-gradient residuals near 10^{-6} versus a 10^{-8} threshold. Three min-norm solves fail, leaving 97 images.

The least-core methods assign negative marker credit more than twice as often as Shapley, including among images where adding the marker patches reduces the target logit (Table 13 and Figure 7). ELC and the two LP-based methods give similar results. This supports the contrast between least-core and Shapley attribution in this task, without a separate advantage for ELC's selector.

Table 13: Marker attribution with one sampling seed. Credit is the sum $x(D)$ in logit units; negative rate is the fraction with $x(D) < 0$. The last column restricts this rate to images where adding D to its complement lowers the target logit: 74 images, or 72 among successful LP + min-norm runs.

Method	Images	Mean $x(D)$	Negative rate	Negative on harmful
Shapley	100	$+0.294$	31.0%	37.8%
LP	100	-0.259	67.0%	77.0%
LP + min-norm	97	-0.250	67.0%	76.4%
ELC	100	-0.257	67.0%	77.0%

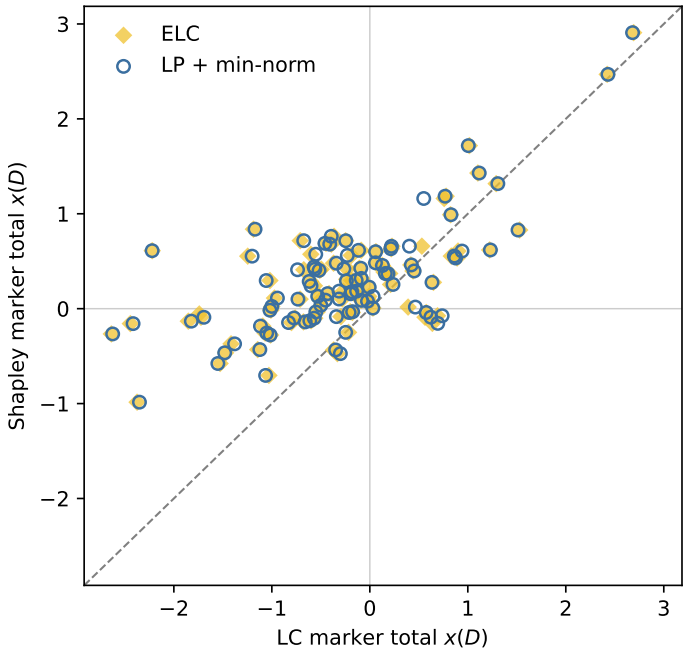

Figure 7: Marker attribution: Shapley versus least-core marker credit $x(D)$, in logit units. Each point represents one image: the horizontal coordinate is ELC (yellow diamonds, 100 images) or LP + min-norm (blue open circles, 97 successful solves), and the vertical coordinate is the corresponding Shapley estimate. The dashed line denotes equal credit; gray lines mark zero. The two least-core series largely overlap.