How Jailbreak Attacks Inform LLM Safety Alignment: A Defender-Centric, Shapley-Based Evaluation of Jailbreak Contributions

Anonymous authors
Paper under double-blind review

Abstract

Jailbreak attacks on large language models are usually evaluated by attacker-centric metrics such as attack success rate (ASR), yet an attack that breaks a model is not necessarily useful for improving its safety. We propose a defender-centric view of jailbreak evaluation, where attacks are evaluated by the downstream safety improvements they enable when used as red-teaming data for safety training. Building on this view, we introduce **A-MESS** (**Minimal Effective Attack-Subset Selection**), a setting-agnostic framework for attributing and selecting jailbreak attacks from black-box subset utility observations. A-MESS estimates **AttackSHAP**, a Shapley-based score for attributing marginal utility to individual attacks, and selects compact attack subsets under user-specified budgets via greedy or surrogate-based optimization. Under limited queries, A-MESS-ACQUISITION balances coalition-size coverage with model disagreement, accelerating early AttackSHAP estimation and reducing subset-selection gaps relative to baseline acquisition methods on controlled landscapes. Across controlled and real LLM safety settings, we find that ASR rankings are weakly aligned with defender-centric utility and that directly optimizing subsets yields stronger safety utility than attacker-centric or attribution-only selection. These results suggest evaluating jailbreak attacks as resources for improving safety, not only as tools for breaking models.

1 Introduction

Jailbreak attacks have become a central tool for stress-testing the safety of large language models (LLMs) Carlini et al. (2023); Wei et al. (2023a); Jain et al. (2023); Yi et al. (2024); Chu et al. (2025b). By constructing adversarial prompts Zou et al. (2023); Zhu et al. (2024b); Liao & Sun (2024); Zhou et al. (2025b); Beyer et al. (2026) or interaction strategies Chao et al. (2025); Ren et al. (2025); Andriushchenko & Flammarion (2025); Huang et al. (2026), these attacks expose failure modes where a model's safety alignment can be bypassed. Existing jailbreak research Chu et al. (2025a); Ran et al. (2024) therefore commonly evaluates attacks using attacker-centric metrics such as attack success rate (ASR), computational cost, universality, and transferability. These metrics measure how effectively and practically an attack can compromise a model from the perspective of an attacker.

For LLM safety researchers, however, this perspective is incomplete. A jailbreak attack is useful for safety research not merely because it breaks an undefended model, but because it can reveal weaknesses that will help improve a defended one Yu et al. (2024); Mo et al. (2024); Fu et al. (2025). By shifting from *attack success* to *defense utility*, this motivates a defender-centric view of jailbreak evaluation: attacks should be valued by their downstream safety utility contribution when used as red-teaming resources in defense pipelines. Under this view, attack success is not the final objective, but an input to a broader safety-improvement process.

This shift exposes a practical decision problem. ***Which jailbreak attack(s) matter most when "matter" means improving downstream safety alignment, rather than maximizing attack success?*** This question cannot be answered by standalone attack-success metrics alone. If ASR were a reliable proxy for defense utility contribution, ranking attacks by no-defense success would already identify the most useful attacks for improving safety. However, the value of an attack may depend on the specific safety setting context, including the defense pipeline, evaluation benchmark, utility definition,

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.

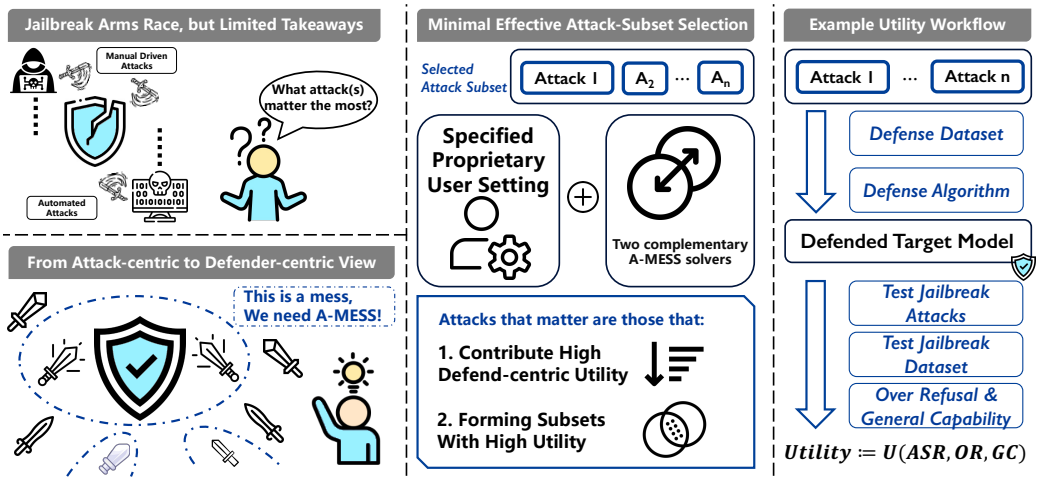


Figure 1: Conventional jailbreak research often follows an attacker-centric arms race, where attacks are primarily judged by how effectively they break a target model. A-MESS instead starts from a defender-centric question: which jailbreak attack(s) matter most for improving safety alignment under a specific, potentially proprietary user setting? Given setting-specific utility observations, A-MESS supports two complementary tasks: identifying attacks with high equitable contribution via AttackSHAP, and selecting compact attack subsets with high downstream utility.

and even the other attacks selected with it for defense at the same time. Thus, the setting is highly heterogeneous, and even potentially proprietary. We therefore model each setting by a black-box utility function $v_\theta(S)$, which measures the safety utility of an attack subset S under a user-specified setting θ. Our goal is not to find a universally best jailbreak attack, but to provide a setting-agnostic framework that can be instantiated wherever such subset utilities can be observed.

This decision problem formulation leads to two related but distinct tasks, as illustrated in Figure 1. The first is attribution: how much credit should each attack receive for contributing to downstream utility? We address this with **AttackSHAP**, a Shapley-based attribution score that treats jailbreak attacks as players, attack subsets S as coalitions, and $v_\theta(S)$ as the coalition value. However, directly employing exact Shapley value computation requires the true utility landscape, which is prohibitively expensive to obtain. To make AttackSHAP practical, we seek accurate estimates under a limited coalition acquisition budget, which further raises a key question to be answered later: *which coalitions should receive utility evaluations?* The second task is subset selection: for a target subset size k, which attack subset yields the highest defender utility? These tasks are not equivalent, since attacks with high individual contribution do not necessarily compose the highest-utility subset.

To address these tasks, we propose **A-MESS** (**Minimal Effective Attack-Subset Selection**), a defender-centric framework for both attack attribution and budgeted attack-subset selection tasks. A-MESS provides two complementary solvers. A-MESS-SURROGATE leverages a surrogate model to learn an approximate utility landscape from sampled subset evaluations, enabling efficient AttackSHAP estimation and multi-budget subset search under a fixed evaluation budget. For the surrogate solver, A-MESS-ACQUISITION selects which coalitions to evaluate by combining coalition-size coverage with within-size model disagreement. A-MESS-GREEDY instead directly queries true marginal utilities when a practitioner needs one subset of a specified size k.

Our contributions are threefold:

- We reframe jailbreak attack evaluation from attacker-centric success to defender-centric utility, and empirically test whether ASR predicts downstream safety utility.
- We introduce **AttackSHAP**, a Shapley-based attribution score for assigning setting-specific utility contribution to individual jailbreak attacks across subset contexts.
- We formulate budgeted attack-subset selection and propose **A-MESS** (**Minimal Effective Attack-Subset Selection**), a setting-agnostic framework with greedy and surrogate-based solvers for selecting compact high-utility attack subsets from black-box utility observations. Empirical results show that A-MESS yields subsets with higher defender-centric utility than attacker-centric or attribution-only selection.

2 PRELIMINARIES AND PROBLEM SETUP

We first formalize the problem discussed above. Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ denote the candidate attack universe, where each a_i is an attack algorithm, or equivalently attack-generated data source. As summarized in Figure 1, a safety setting θ specifies how one attack subset is converted into a defended system and how that system is evaluated. It may include the target model, defense procedure, evaluation benchmarks, computation budget, and utility definition. For any subset $S \subseteq \mathcal{A}$, we define a defender-centric utility $v_\theta : 2^{\mathcal{A}} \to \mathbb{R}$, where $v_\theta(S)$ is the utility obtained after applying the defense workflow with attacks in S. The utility may measure jailbreak ASR reduction, or combine it with other objectives such as over-refusal penalty. A-MESS does not assume access to the internal details of θ, it only assumes that a practitioner can query $v_\theta(S)$ for selected subsets. Throughout the paper, we instantiate v_θ in both real LLM safety settings and controlled synthetic utility landscapes. In each case, utilities are computed according to the corresponding setting, and normalized to a common $[0, 1]$ scale. Detailed utility settings are provided in the experimental sections and Appendix.

In this formulation, a utility query $v_\theta(S)$ corresponds to executing the defender's evaluation workflow for one attack subset: constructing defense data or prompts from S, applying the defense procedure, and evaluating the resulting defended system under the setting-specific safety metrics. This black-box view is important because many realistic safety pipelines stay proprietary, as model providers tend not to release the entire pipeline. Attacker-centric quantities such as no-defense ASR can be used as baselines, but they do not explicitly query $v_\theta(S)$.

Individual attack attribution. A central goal is to quantify how much each attack contributes to the safety utility. This cannot be captured by standalone attack metrics since they ignore subset context: Two attacks may be redundant if they expose the same failure mode, so adding one after the other yields little additional utility. Conversely, two individually modest attacks may be complementary if they cover different weaknesses and jointly improve the defended system. AttackSHAP is designed to average such marginal contributions across subset contexts. We therefore view attacks as players in a cooperative game, attack subsets $S \subseteq \mathcal{A}$ as coalitions and $v_\theta(S)$ as the coalition value. The AttackSHAP value of attack a_i is then:

$$\phi_i^\theta = \sum_{S \subseteq \mathcal{A} \setminus \{a_i\}} \frac{|S|!(n - |S| - 1)!}{n!} \left[v_\theta(S \cup \{a_i\}) - v_\theta(S) \right]. \tag{1}$$

This is the Shapley value Shapley et al. (1953); Dubey (1975); Ghorbani & Zou (2019) of a_i in the utility game induced by θ, and quantifies the average marginal utility of adding a_i across subset contexts. We compare AttackSHAP with two boundary marginal baselines: $\mathrm{AOO}_i^\theta = v_\theta(\{a_i\}) - v_\theta(\emptyset)$, which measures empty-context gain, and $\mathrm{LOO}_i^\theta = v_\theta(\mathcal{A}) - v_\theta(\mathcal{A} \setminus \{a_i\})$, which measures full-context loss (Koh & Liang, 2017; Ghorbani & Zou, 2019). We compare estimated AttackSHAP against reference scores using standard regression and correlation metrics.

Budgeted subset selection. Attribution ranks individual attacks, but LLM safety researchers often need a subset-level decision in practice. Given a budget k, the oracle objective is to select the size-k subset with the highest safety utility

$$S_k^\star = \arg \max_{S \subseteq \mathcal{A},\, |S| = k} v_\theta(S). \tag{2}$$

Here $S_k^\star$ denotes the oracle optimal subset under setting θ. This fixed-size formulation captures the common case where a defender can only use k attacks for training or evaluation. In practice, finding $S_k^\star$ exactly is infeasible for large n, because it requires evaluating $\binom{n}{k}$ candidate subsets. We use $S_k^\star$ as an oracle upper bound when the full utility landscape is available, and introduce practical A-MESS solvers for approximating high-utility subsets $S_k(\text{A-MESS})$ in Section 3.

The acquisition bottleneck. Exact AttackSHAP estimation and exhaustive search over size-k subsets both require prohibitively many true utility evaluations. With a limited coalition acquisition budget, the bottleneck for A-MESS-SURROGATE is deciding which coalitions merit true utility evaluations. Coalition acquisition should therefore balance the training set across subset sizes and prioritize informative coalitions that reduce uncertainty in the learned utility model (Section 3.1).

3 METHOD

3.1 COALITION UTILITY ACQUISITION AND SURROGATE LEARNING

Coalition utility acquisition. We distinguish two passive strategies for constructing the surrogate training set. UNIFORM-COALITION samples uniformly without replacement from all 2^n coalitions. The induced probability of cardinality k is $\binom{n}{k}/2^n$, so most queries concentrate near the middle sizes. UNIFORM-SIZE instead balances query counts across non-exhausted cardinalities and samples uniformly without replacement within each size. This provides broader size coverage, but does not use observed utilities to choose which coalitions to query.

A-MESS-ACQUISITION retains this size coverage and applys query-by-committee Seung et al. (1992) to prioritize informative candidates within each size. It starts from the same Uniform-size prefix, then selects coalitions with the largest prediction disagreement among guide models trained on the observed data. Algorithm 1 summarizes the procedure; training settings and tie-breaking are specified in Appendix K. Writing c_k for acquired size-k counts, $\mathcal{U}$ for remaining candidates, and $d(S)$ for committee prediction variance, each batch addition follows

Algorithm 1 A-MESS-ACQUISITION

Require: Utility function v, acquisition budget M, acquisition batch size b, committee size J
$D \leftarrow$ queried Uniform-size warm-start pairs
while $|D| < M$ **do**
  Train guide MLPs $f_1, \ldots, f_J$ on D
  $\mathcal{U} \leftarrow$ all unqueried coalitions
  $S \in \mathcal{U}$: a coalition with unqueried utility
  $d(S) \leftarrow \mathrm{Var}_j[f_j(S)], S \in \mathcal{U}$
  Initialize size counts c_k from D
  $B \leftarrow \emptyset$ *// acquisition batch*
  for $i = 1, \ldots, \min(b, M - |D|)$ **do**
    Select $(k^\star, S^\star)$ by Eq. 3
    // Balance sizes first, then maximize disagreement.
    $B \leftarrow B \cup \{S^\star\}; \mathcal{U} \leftarrow \mathcal{U} \setminus \{S^\star\}$
    $c_{k^\star} \leftarrow c_{k^\star} + 1$
  end for
  Query v on B; append pairs to D
end while
return D *// acquired training data*

$$k^\star \in \arg\min_{k:\, \exists S \in \mathcal{U},\, |S|=k} c_k, \qquad S^\star \in \arg\max_{S \in \mathcal{U},\, |S|=k^\star} d(S). \tag{3}$$

Surrogate fitting. A-MESS-SURROGATE represents coalition T by its binary membership vector $x_T \in \{0,1\}^n$. Given M queried pairs $\mathcal{D}_{\text{train}} = \{(x_{T_j}, v_\theta(T_j))\}_{j=1}^M$, it fits $\hat{v}_\theta$ by minimizing MSE. The fitted model supports AttackSHAP estimation and multi-budget subset search without additional true utility queries. Only queried coalition utilities are available during acquisition and surrogate fitting. Held-out test utilities are used solely for evaluation.

3.2 ATTACK ATTRIBUTION AND RANKING

A-MESS uses AttackSHAP to estimate the contribution of each attack. When the true utility landscape is available, we compute reference scores $\phi_i^\theta = \phi_i(v_\theta)$, either exactly in controlled synthetic settings or approximately through sampled coalitions. When using a surrogate model, A-MESS-SURROGATE estimates $\hat{\phi}_i^\theta = \phi_i(\hat{v}_\theta)$, replacing true utility evaluations with surrogate predictions. Thus, after the initial utilities are queried to fit the surrogate model, additional coalitions required by Shapley estimation are evaluated by $\hat{v}_\theta$ rather than by re-running the full defense workflow.

Attribution scores can also be used for top-k selection by ranking attacks and returning the highest-scoring k elements. We consider rankings induced by ASR, AttackSHAP, LOO Koh & Liang (2017), and AOO. This allows us to separate two questions: whether utility can be accurately attributed to each individual attack, and whether selecting attacks by individual importance is sufficient for subset-level utility maximization.

3.3 BUDGETED SUBSET SELECTION

Beyond ranking individual attacks, A-MESS directly addresses the budgeted subset selection problem. Given a subset size k, the goal is to return a high-utility subset $S_k(m)$ of selection method m under the current setting. We consider two complementary solvers.

A-MESS-GREEDY. The greedy solver directly queries v_θ and greedily selects attacks by marginal gain, making it suitable for single-budget selection when true utility queries are affordable. It starts from $S_0 = \emptyset$ and iteratively adds the attack with the largest true marginal utility gain:

$$a_t^\star = \arg\max_{a \in \mathcal{A} \setminus S_t} \big[v_\theta(S_t \cup \{a\}) - v_\theta(S_t) \big], \tag{4}$$

$$S_{t+1} = S_t \cup \{a_t^\star\}. \tag{5}$$

After k steps, it returns $S_k(\text{A-MESS-GREEDY}) = S_k$. This solver is appropriate when the practitioner only needs one subset under a fixed budget and can afford adaptive true utility queries. Its query cost is $\sum_{t=0}^{k-1}(n-t) = nk - k(k-1)/2$ for a size-k subset, excluding cached evaluations.

A-MESS-SURROGATE. The surrogate solver first learns $\hat{v}_\theta$ from subset evaluations sampled by A-MESS-ACQUISITION, and then searches in the learned utility landscape. For a target budget k, it returns:

$$S_k(\text{A-MESS-SURROGATE}) = \arg\max_{S \subseteq \mathcal{A},\, |S|=k} \hat{v}_\theta(S). \tag{6}$$

3.4 SAFETY-SETTING INSTANTIATIONS

A-MESS only requires black-box access to $v_\theta(S)$, but in real LLM experiments we instantiate three safety settings θ with two defense pipelines: adversarial training Mazeika et al. (2024) and in-context defense Wei et al. (2023b) (ICD). We distinguish defense jailbreak data from test data: the former is collected for the selected attack subset S and used to build the defended system, while the latter is used to evaluate $v_\theta(S)$. The resulting defended system is evaluated on test jailbreak data, over-refusal safe requests, and general-capability questions, as illustrated in Figure 1.

4 EXPERIMENTS

4.1 EVALUATED ASPECTS AND METRICS

We evaluate A-MESS along three aspects. First, we test whether attacker-centric success is aligned with defender-centric utility by comparing the no-defense ASR ranking with the singleton utility ranking. We report Spearman's ρ, Kendall's τ, and representative rank shifts. Second, we evaluate whether AttackSHAP can be estimated under limited utility queries. For synthetic landscapes, we compare estimated AttackSHAP against exact references using regression and rank-correlation metrics. For real LLM utilities, all 1,024 subsets are evaluated; the surrogate sees only its designated training labels, and prediction is measured on held-out subsets. Third, we evaluate budgeted subset selection using raw subset utility $v_\theta(S_k(m))$ and Normalized Gain:

$$\text{NG}_k(m) = \frac{v_\theta(S_k(m)) - v_\theta(S_k(\text{BASE}))}{v_\theta(S_k^\star) - v_\theta(S_k(\text{BASE}))}. \tag{7}$$

Here $S_k(\text{BASE})$ is the average random size-k subset and $S_k^\star$ is the oracle size-k subset when available. Thus, $NG_k(m)$ measures the fraction of the oracle improvement gap recovered by method m.

4.2 EXPERIMENTAL SETUP

Synthetic utility landscapes. We use synthetic utility landscapes to model defender-side utilities over attack subsets. These experiments do not synthesize jailbreak attacks; instead, each task defines a bounded utility oracle $v(S) \in [0, 1]$ over a fixed attack universe $\mathcal{A}$. Each task contains $n = 20$ candidate attacks, allowing exact enumeration of all 2^{20} subset utilities and exact reference AttackSHAP values. We instantiate 90 utility landscapes from scalar, vector, and random generation policies. Across policies, the construction is designed to capture patterns we expect in real defense utilities, including diminishing gains, redundant attacks, and complementary attack combinations. Figure 4 illustrates a representative synthetic utility landscape.

Real LLM safety settings. Following the workflow in Figure 1, we report Qwen2.5-7B with adversarial supervised fine-tuning (SFT) in the main text. The appendix reports Ministral-3-8B with in-context defense (ICD) and adversarial SFT. Selected JailbreakBench attacks provide the defense data or context. Across settings, the attack universe $\mathcal{A}$ contains 10 attack categories instantiated on

Table 1: Compared methods. Utility budget denotes the number of true subset-utility evaluations $v(S)$, excluding attacker-centric quantities such as ASR. M is the utility budget specified by each concrete method.

Method	Utility Budget	Description
Attribution methods		
ASR	0	Attacker-centric ranking by attack success rate
AOO	$n+1$	Empty-context marginal utility gain
LOO	$n+1$	Full-context marginal utility loss
STANDARD SHAP$_v$	2^n	Reference Shapley attribution from true utilities
ATTACKSHAP$_{\hat{v}}$	M	Shapley attribution from learned surrogate utilities
SHAP BASELINES	M	Standard Shapley value estimation baselines, including DataSHAP Ghorbani & Zou (2019), SVA Zhang et al. (2023), SamplingSHAP@K Kariyappa et al. (2024)
Subset selection methods		
RANDOM	0	Random size-k subset
TOPK$(\cdot)$	score-dependent	Select top k attacks by an attribution score listed above
A-MESS-GREEDY	$nk - k(k-1)/2$	Greedy selection using true marginal utility
A-MESS-SURROGATE	M	Search over the learned utility landscape
ORACLESEARCH$_v$	$\binom{n}{k}$	Exhaustive true-utility search; not feasible and acts as upper bound only

Table 2: Illustrative examples of Qwen2.5-7B + SFT setting rank changes: unchanged, lower, and higher singleton utility rank. ASR Rank uses no-defense JBB ASR. Utility Rank uses all ten singleton defenses. $\Delta\text{Rank} = \text{ASR Rank} - \text{Utility Rank}$.

Attack	ASR Rank	No-defense ASR	Utility Rank	Utility	ΔRank
`Roleplay`	1	0.600	1	0.200	+0
`Optimized Suffix`	6	0.340	10	0.125	-4
`Low Resource Language`	8	0.090	5	0.153	+3
Spearman $\rho = 0.188$, Kendall $\tau_b = 0.111$.					

JailbreakBench and SorryBench, covering template-based Wei et al. (2023b); Shen et al. (2024); Yoosuf et al. (2025); Zhu et al. (2024a), strategy-based Andriushchenko & Flammarion (2025); Huang et al. (2026); Luo et al. (2026), and learning-based Zou et al. (2023); Zhou et al. (2025b); Zhu et al. (2024b) jailbreaks. The two defense algorithms, adversarial training and in-context defense, instantiate different ways in which an attack subset S changes the defended system. For subset selection, we evaluate budgets $k \in \{2, 4\}$.

Shared real-setting utility. All real LLM experiments use the same defender-centric utility, so in these settings $v_\theta(S) = U(S)$ for singleton ranking, held-out subset utility prediction, and budgeted subset selection. Evaluation covers JailbreakBench Chao et al. (2024), SorryBench Xie et al. (2024), safe XSTest requests for over-refusal evaluation Röttger et al. (2024), and tinyMMLU-100 Polo et al. (2024). For test attack j, let $a_j(S) = \frac{1}{2}\text{ASR}_{j,\text{JBB}}(S) + \frac{1}{2}\text{ASR}_{j,\text{Sorry}}(S)$ denote its equal-weighted benchmark ASR. Let $n = |\mathcal{A}|$ be the number of attack categories, with $n = 10$ in our real LLM experiments. The ASR score $s_{\text{ASR}}(S)$ measures average relative risk reduction, and the ORR score $s_{\text{ORR}}(S)$ measures safe-request non-refusal:

$$s_{\text{ASR}}(S) = \frac{1}{n}\sum_{j=1}^{n} \text{clip}\left(\frac{a_j(\emptyset) - a_j(S)}{a_j(\emptyset)}, 0, 1\right). \tag{8}$$

$$s_{\text{ORR}}(S) = 1 - \text{ORR}(S). \tag{9}$$

$$U(S) = 0.9\, s_{\text{ASR}}(S) + 0.05\, s_{\text{ORR}}(S) + 0.05\, \text{Acc}(S). \tag{10}$$

We combine benchmarks within each attack before computing its relative reduction, then average over attacks. All ten attacks use both benchmarks with equal weights, and all observed empty-baseline denominators are positive. ORR is measured on 100 safe XSTest requests; Acc is accuracy on 100 tinyMMLU questions. Clipping assigns zero reduction when ASR is unchanged or increases, so small baseline ASRs can amplify proportional changes. Because the full JBB evaluation includes the three defense behaviors, these results describe full-benchmark performance rather than generalization to a strictly disjoint holdout. Full implementation details, attack taxonomy, and hyperparameters are provided in Appendices D–F.

Compared methods. Table 1 summarizes the compared methods. We separate methods that produce attribution scores from methods that directly return a size-k subset. Attribution methods can be converted into subset selectors by ranking attacks and taking the top k, while selection methods optimize the subset-level decision more directly.

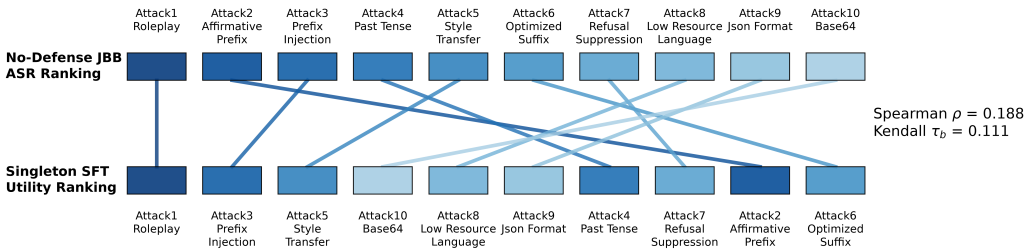


Figure 2: Attack success and defense utility yield different rankings for Qwen2.5-7B + SFT. Attacks are ranked by no-defense JBB ASR in the upper row and by singleton defense utility in the lower row. Lines connect the same attack across rankings.

4.3 From Attack Success to Defense Utility

We compare attack success with defender-centric utility for Qwen2.5-7B + adversarial SFT. The attacker-centric ranking uses each attack's no-defense JailbreakBench ASR only. The defender-centric ranking uses the shared utility in Eq. (10) after SFT on one attack category, evaluated against all ten test attack categories. Attack IDs in the ranking figures and subset-selection tables follow each model's descending no-defense JBB ASR order.

Figure 2 shows limited rank agreement: Spearman $\rho = 0.188$ and Kendall $\tau_b = 0.111$ across ten attacks. Table 2 illustrates unchanged, lower, and higher utility ranks rather than selecting the three highest-ASR attacks. Roleplay remains first; Optimized Suffix moves from sixth to tenth; Low Resource Language moves from eighth to fifth. Thus JBB attack success alone does not recover singleton utility ordering for this relative-risk objective. Appendix G repeats the full ranking comparison for Ministral + ICD and Ministral + SFT.

Table 3: Qwen2.5-7B + SFT utility prediction on a fixed 500-subset pool. Mean ± sample std over model seeds 2026–2030; each row uses one fixed train/test split.

n_{train}:n_{test}	MAE ↓	R^2 ↑	Pearson ↑	Spearman ↑
100:400	0.065± 0.005	0.833± 0.024	0.918± 0.013	0.920± 0.010
200:300	0.056± 0.007	0.876± 0.029	0.940± 0.013	0.941± 0.010
300:200	0.046± 0.002	0.911± 0.007	0.959± 0.004	0.952± 0.002
400:100	0.037± 0.003	0.952± 0.007	0.977± 0.004	0.976± 0.002

Table 4: Synthetic subset selection results. We report Normalized Gain NG_k (%) averaged over 90 utility landscapes, where higher values indicate closer recovery of the oracle subset utility.

Method	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$
Attribution Baselines					
TopK-AOO$_v$	**100.00**	53.12	46.66	37.76	31.52
TopK-LOO$_v$	33.99	36.48	44.79	57.33	63.80
TopK-ϕ_v	65.39	61.10	67.03	68.51	69.28
TopK-$\hat{\phi}$	59.66	57.37	62.94	65.01	71.07
A-MESS Solvers					
A-MESS-Greedy	**100.00**	**85.47**	**83.36**	**83.20**	82.69
A-MESS-Surrogate	59.78	73.18	74.48	79.79	**84.52**

4.4 Query-Efficient Utility Learning and AttackSHAP Estimation

We first compare which coalition utilities to acquire, then evaluate surrogate prediction and AttackSHAP estimation under limited query budgets.

Coalition utility acquisition. We compare Uniform-coalition, Uniform-size, and A-MESS-Acquisition on all 90 synthetic landscapes with the same MLP and training protocol. Uniform-size and A-MESS-Acquisition share their first 100 queries; full evaluation details appear in Appendix K. Figure 3 shows that A-MESS-Acquisition reduces Shapley prediction error faster in the early acquisition stage and reaches lower subset selection gaps as the query budget grows. The sustained benefit is better downstream subset selection, showing the value of using model disagreement beyond size coverage alone.

Surrogate utility learning and AttackSHAP estimation. We evaluate whether A-MESS-Surrogate can estimate AttackSHAP with limited utility queries. On synthetic landscapes, the full 2^{20} utility table provides oracle AttackSHAP references. Appendix Figure 7 shows that the learned utility model reaches high R^2 in the high-budget regime. More importantly, the AttackSHAP estimates match or outperform strong existing Shapley estimators on both full-vector ϕ prediction and top-k coverage, while requiring only hundreds of true utility queries rather than thousands or tens of thousands. A-MESS reaches a strong accuracy efficiency tradeoff for standard Shapley estimation.

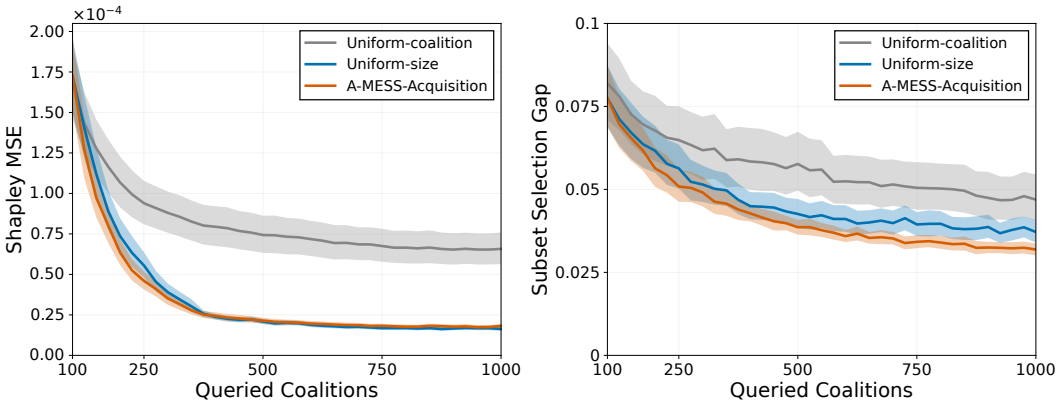


Figure 3: Learning useful coalition utilities under limited queries. A-MESS-ACQUISITION accelerates early Shapley error reduction and achieves lower subset selection gaps than passive acquisition. Lines show means across 90 landscapes, shading indicates pointwise 95% confidence intervals.

Table 5: Qwen2.5-7B + SFT defense utility under attack-subset selection with $k = 2$ and $k = 4$. We compare the selected subsets, their observed utility, and the gain over the no-defense baseline (ΔU). Bold values indicate the best utility at each subset size.

Method	Selected subset	Utility↑	ΔU↑	Method	Selected subset	Utility↑	ΔU↑
NO DEFENSE	–	0.0835	0.0000	NO DEFENSE	–	0.0835	0.0000
Baselines with budget $k = 2$				*Baselines with budget $k = 4$*			
RANDOM	20 random subsets	0.2710	0.1875	RANDOM	20 random subsets	0.6111	0.5276
TOPK-ASR	Attack_1, Attack_6	0.2640	0.1805	TOPK-ASR	Attack_1, Attack_2, Attack_3, Attack_6	0.6256	0.5421
TOPK-AOO	Attack_1, Attack_3	**0.3880**	**0.3045**	TOPK-AOO	Attack_1, Attack_3, Attack_5, Attack_{10}	0.6606	0.5771
TOPK-LOO	Attack_7, Attack_9	0.1759	0.0924	TOPK-LOO	Attack_2, Attack_7, Attack_8, Attack_9	0.4842	0.4007
A-MESS Solvers with budget $k = 2$				*A-MESS Solvers with budget $k = 4$*			
A-MESS-GREEDY	Attack_1, Attack_3	**0.3880**	**0.3045**	A-MESS-GREEDY	Attack_1, Attack_3, Attack_4, Attack_5	0.7170	0.6335
A-MESS-SURROGATE	Attack_1, Attack_3	**0.3880**	**0.3045**	A-MESS-SURROGATE	Attack_3, Attack_4, Attack_5, Attack_9	**0.8542**	**0.7707**

For real Qwen2.5-7B + SFT utilities, Table 3 shows that MAE decreases from 0.065 to 0.037 and R^2 increases from 0.833 to 0.952 across the evaluated training budgets. Appendix L details the fixed-pool splits and seed aggregation; Tables 8 and 9 report the corresponding Ministral results.

4.5 MINIMAL EFFECTIVE SUBSET SELECTION

We finally evaluate whether attack attribution directly yields the best subset, and whether A-MESS can improve subset-level utility. Table 4 reports Normalized Gain on 90 synthetic utility landscapes. Attribution-based top-k selection improves over random selection, but remains clearly below direct subset optimization. Among attribution baselines, TOPK-ϕ_v and TOPK-$\hat{\phi}$ are the strongest, reaching roughly 60–71% Normalized Gain for nontrivial budgets. In contrast, A-MESS-GREEDY recovers at least 82.69% of the oracle improvement gap for $k > 1$, while A-MESS-SURROGATE improves as the budget grows and reaches 84.52% at $k = 9$. These results show that ranking attacks by individual contribution is not sufficient, and explicitly optimizing the subset can recover more oracle gain.

Table 5 extends this comparison to Qwen2.5-7B with adversarial SFT. At $k = 2$, both A-MESS solvers match the strongest attribution baseline; at $k = 4$, they outperform the ranking-based alternatives. The surrogate achieves the highest utility among the compared methods at the larger budget, reinforcing the value of searching over attack combinations.

These results connect the defender-centric evaluation perspective to a practical selection decision: attacks should be chosen for their joint defense utility, rather than standalone attack success. A-MESS-SURROGATE uses the utilities collected by A-MESS-ACQUISITION to support this subset-level search across budgets. The displayed surrogate run was selected from five model seeds, and query costs differ across methods; the comparison therefore illustrates attainable utility rather than a matched-cost advantage. Appendix I details the protocol and the corresponding Ministral results. Appendix J reports the Qwen2.5-7B + SFT subset-selection results under two alternative utilities.

5 DISCUSSION

The main contribution of this work is not merely a numerical improvement, but a shift in how jailbreak attacks should be interpreted in safety research. Much of jailbreak research has been driven by an arms race. However, a first-principles view of doing research should start from the very final goal Lu et al. (2025). Applied to jailbreak research, this view suggests that attacks should be valued not only by how often they break an undefended model, but by how much they improve downstream safety when used as red-teaming resources. Our results make this distinction concrete: ASR and singleton defense utility are weakly aligned in our real settings, and attacks with high individual attribution do not necessarily form the best subset. This reframes jailbreak attacks as defense resources whose value depends on the downstream safety objective.

This perspective also motivates the black-box, setting-agnostic design of A-MESS. Safety settings may be largely heterogeneous Chu et al. (2026a); Zhou et al. (2025a;c); Leng et al. (2026) and proprietary, and many production pipelines are only partially observable. Rather than seeking a universally best subset, as a setting-agnostic framework A-MESS requires only utility observations $v_\theta(S)$, allowing practitioners to instantiate the framework under their own safety objectives. This does not solve the broader transparency gap, but it provides a practical interface for connecting jailbreak evaluation to measurable safety improvement.

6 RELATED WORK

Jailbreak attacks and safety defenses. Prior work has proposed diverse jailbreak attacks against aligned LLMs, including template-based Wei et al. (2023b); Shen et al. (2024); Yoosuf et al. (2025); Zhu et al. (2024a), strategy-based Andriushchenko & Flammarion (2025); Huang et al. (2026); Luo et al. (2026), and learning-based attacks Zou et al. (2023); Zhou et al. (2025b); Zhu et al. (2024b). These attacks are typically evaluated by attacker-centric metrics such as ASR, transferability, universality, and cost Chu et al. (2026b). In parallel, LLM safety work has studied refusal training, in-context defenses, guardrails, and over-refusal-aware evaluation Yu et al. (2024); Mo et al. (2024); Fu et al. (2025); Röttger et al. (2024); Zhang et al. (2025); Cui et al. (2025); Pan et al. (2025). These defenses show that attack data can improve safety, but rarely ask which attacks are most useful for downstream safety alignment. A-MESS bridges this gap by treating jailbreak attacks as red-teaming resources and evaluating them through setting-specific defender-side utility.

Game-theoretic valuation and subset selection. Shapley values have been widely used to attribute machine-learning utility to data points, features, or training examples Shapley et al. (1953); Dubey (1975); Ghorbani & Zou (2019), with later work studying more efficient Shapley approximation and top-k identification Zhang et al. (2023); Kariyappa et al. (2024). A-MESS differs in both the object being valued and the decision it supports: the players are jailbreak attacks, the coalition value is downstream defender-side utility under a safety pipeline, and attribution is paired with budgeted subset selection rather than treated as the final goal.

Active utility acquisition. Active learning uses committee disagreement to select informative queries Seung et al. (1992); Krogh & Vedelsby (1994). Shapley approximation instead reduces coalition evaluations through Monte Carlo or stratified sampling Castro et al. (2009); Burgess & Chapman (2021); Zhang et al. (2023). A-MESS-ACQUISITION differs in its goal: it learns a reusable utility landscape for A-MESS-SURROGATE, so the same queried coalitions support both AttackSHAP and multi-budget subset search.

7 CONCLUSION

We presented A-MESS, a defender-centric framework for query-efficient utility acquisition, attack attribution, and compact subset selection. On controlled landscapes, A-MESS-ACQUISITION accelerates early AttackSHAP estimation and reduces subset-selection gaps. In real settings, high-ASR or high-contribution attacks need not form the best defense subsets. These findings suggest that jailbreak evaluation should ask not only which attacks break models, but also which utility observations and attack combinations most effectively support safer defended systems.

AI USE STATEMENT

In this work, we used generative AI tools for formulate mathematical claims, assist in the writing of proofs, design or provide feedback on research methodology or experiments, interpret results. We have not used generative AI tools for provide critical ingredients for proving mathematical claims, propose or refine hypotheses, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, help develop theoretical models or conceptual frameworks, generate synthetic data sets. Assist with translation are not applicable to this work. Additionally, we used generative AI tools for suggest experimental parameters, brainstorming, edit a research paper to improve readability. We have reviewed all AI-assisted work. We checked AI-assisted mathematical claims and proof drafts against our own derivations, and compared suggestions about experimental design and result interpretation with the implemented protocols and observed results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies existing jailbreak attacks as red-teaming resources for improving LLM defenses. It does not propose a new attack-generation method. Although A-MESS measures downstream defense utility rather than attack success, the analysis of jailbreak behavior remains potentially dual-use, and associated model outputs are not going to be publicly released.

REPRODUCIBILITY STATEMENT

Reproduction details are provided in Appendices B and D for the synthetic and real LLM settings, and in Appendices F, K, and L for the baselines, acquisition protocol, and surrogate training.

REFERENCES

Maksym Andriushchenko and Nicolas Flammarion. Does refusal training in LLMs generalize to the past tense? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=aJUuere4fM.

Tim Beyer, Yan Scholten, Leo Schwinn, and Stephan Günnemann. Sampling-aware adversarial attacks against large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vBmRQHW7en.

Mark A. Burgess and Archie C. Chapman. Approximating the shapley value using stratified empirical bernstein sampling. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 73–81. International Joint Conferences on Artificial Intelligence Organization, 2021. doi: 10.24963/ijcai.2021/11. URL https://doi.org/10.24963/ijcai.2021/11.

Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramer, and Ludwig Schmidt. Are aligned neural networks adversarially aligned? *Advances in Neural Information Processing Systems*, 36:61478–61500, 2023.

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.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42. IEEE, 2025.

Junjie Chu, Mingjie Li, Ziqing Yang, Ye Leng, Chenhao Lin, Chao Shen, Michael Backes, Yun Shen, and Yang Zhang. Jades: A universal framework for jailbreak assessment via decompositional scoring. *arXiv preprint arXiv:2508.20848*, 2025a.

Junjie Chu, Yugeng Liu, Ziqing Yang, Xinyue Shen, Michael Backes, and Yang Zhang. Jailbreakradar: Comprehensive assessment of jailbreak attacks against llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21538–21566, 2025b.

Junjie Chu, Yiting Qu, Ye Leng, Michael Backes, Yun Shen, Savvas Zannettou, and Yang Zhang. Understanding llm behavior when encountering user-supplied harmful content in harmless tasks, 2026a. URL https://arxiv.org/abs/2603.11914.

Junjie Chu, Xinyue Shen, Ye Leng, Michael Backes, Yun Shen, and Yang Zhang. Benchmark of benchmarks: Unpacking influence and code repository quality in llm safety benchmarks, 2026b. URL https://arxiv.org/abs/2603.04459.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. OR-bench: An over-refusal benchmark for large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=CdFnEu0JZV.

Pradeep Dubey. On the uniqueness of the shapley value. *International Journal of Game Theory*, 4(3): 131–139, 1975.

Shaopeng Fu, Liang Ding, Jingfeng Zhang, and Di Wang. Short-length adversarial training helps llms defend long-length jailbreak attacks: Theoretical and empirical evidence. *arXiv preprint arXiv:2502.04204*, 2025.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.

Xun Huang, Simeng Qin, Xiaoshuang Jia, Ranjie Duan, Huanqian Yan, Zhitao Zeng, Fei Yang, Yang Liu, and Xiaojun Jia. Obscure but effective: Classical chinese jailbreak prompt optimization via bio-inspired search. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=O7fxz7D6vf.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

Sanjay Kariyappa, Leonidas Tsepenekas, Freddy Lécué, and Daniele Magazzeni. Shap@ k: Efficient and probably approximately correct (pac) identification of top-k features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13068–13075, 2024.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7, pp. 231–238. MIT Press, 1994. URL https://proceedings.neurips.cc/paper/1994/hash/b8c37e33defde51cf91e1e03e51657da-Abstract.html.

Ye Leng, Junjie Chu, Mingjie Li, Chenhao Lin, Chao Shen, Michael Backes, Yun Shen, and Yang Zhang. When understanding becomes a risk: Authenticity and safety risks in the emerging image generation paradigm, 2026. URL https://arxiv.org/abs/2603.24079.

Zeyi Liao and Huan Sun. AmpleGCG: Learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed LLMs. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=UfqzXg95I5.

Renzhi Lu, Zonghe Shao, Yuemin Ding, Ruijuan Chen, Dongrui Wu, Housheng Su, Tao Yang, Fumin Zhang, Jun Wang, Yang Shi, et al. Discovery of the reward function for embodied reinforcement learning agents. *Nature Communications*, 16(1):11064, 2025.

Xuan Luo, Yue Wang, Zefeng He, Geng Tu, Jing Li, and Ruifeng Xu. A simple and efficient learning-style prompting for llm jailbreaking. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 2389–2406, 2026.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Yichuan Mo, Yuji Wang, Zeming Wei, and Yisen Wang. Fight back against jailbreaking via prompt adversarial tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=nRdST1qifJ.

Licheng Pan, Yongqi Tong, Xin Zhang, Xiaolu Zhang, Jun Zhou, and Zhixuan Chu. Understanding and mitigating overrefusal in llms from an unveiling perspective of safety decision boundary. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21068–21086, 2025.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: evaluating LLMs with fewer examples. *arXiv preprint arXiv:2402.14992*, 2024.

Delong Ran, Jinyuan Liu, Yichen Gong, Jingyi Zheng, Xinlei He, Tianshuo Cong, and Anyu Wang. Jailbreakeval: An integrated toolkit for evaluating jailbreak attempts against large language models. *arXiv preprint arXiv:2406.09321*, 2024.

Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. Llms know their vulnerabilities: Uncover safety gaps through natural distribution shifts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24763–24785, 2025.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, 2024.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 287–294. ACM, 1992. doi: 10.1145/130385.130417.

Lloyd S Shapley et al. A value for n-person games. 1953.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1671–1685, 2024.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36:80079–80110, 2023a.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023b.

Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, et al. Sorry-bench: Systematically evaluating large language model safety refusal behaviors. *arXiv preprint arXiv:2406.14598*, 2024.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*, 2024.

Shehel Yoosuf, Temoor Ali, Ahmed Lekssays, Mashael AlSabah, and Issa Khalil. Structtransform: A scalable attack surface for safety-aligned large language models. In *European Symposium on Research in Computer Security*, pp. 488–507. Springer, 2025.

Lei Yu, Virginie Do, Karen Hambardzumyan, and Nicola Cancedda. Robust llm safeguarding via refusal feature adversarial training. *arXiv preprint arXiv:2409.20089*, 2024.

Jiayao Zhang, Qiheng Sun, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. Efficient sampling approaches to shapley value approximation. *Proceedings of the ACM on Management of Data*, 1(1):1–24, 2023.

Yuyou Zhang, Miao Li, William Han, Yihang Yao, Zhepeng Cen, and Ding Zhao. Safety is not only about refusal: Reasoning-enhanced fine-tuning for interpretable llm safety. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18727–18746, 2025.

Kaiwen Zhou, Chengzhi Liu, Xuandong Zhao, Anderson Compalas, Dawn Song, and Xin Eric Wang. Multimodal situational safety. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=I9bEi6LNgt.

Yukai Zhou, Jian Lou, Zhijie Huang, Zhan Qin, Sibei Yang, and Wenjie Wang. Don't say no: Jailbreaking llm by suppressing refusal. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25224–25249, 2025b.

Yukai Zhou, Sibei Yang, and Wenjie Wang. Beyond jailbreaks: Revealing stealthier and broader llm security risks stemming from alignment failures. *arXiv preprint arXiv:2506.07402*, 2025c.

Sicheng Zhu, Brandon Amos, Yuandong Tian, Chuan Guo, and Ivan Evtimov. Advprefix: An objective for nuanced llm jailbreaks. *arXiv preprint arXiv:2412.10321*, 2024a.

Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. AutoDAN: Interpretable gradient-based adversarial attacks on large language models. In *First Conference on Language Modeling*, 2024b. URL https://openreview.net/forum?id=INivcBeIDK.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Appendix Overview

This appendix collects the protocols and additional evidence behind the main results. Appendices B and C detail synthetic utility landscapes, their visualization, and raw subset-selection results. Appendices D–F document the real LLM settings, attack categories, and baseline and solver details. Appendices G–I report additional singleton rankings, surrogate predictions, and real subset-selection results. Appendices K and L give the coalition-acquisition protocol, synthetic AttackSHAP results, and real-LLM utility-learning diagnostics.

B Synthetic Utility Landscapes

Purpose. The synthetic experiments are designed to evaluate A-MESS under controlled defender-side utility functions where the full subset utility table is observable. They do not synthesize jailbreak prompts. Instead, each synthetic task defines a set function $v(S)$ over subsets $S \subseteq \mathcal{A}$, which allows us to compute exact AttackSHAP values, oracle size-k subsets, and Normalized Gain.

Task construction. Each synthetic task contains $n = 20$ candidate attacks, so the full utility landscape contains 2^{20} subset values. We instantiate 90 landscapes using three generation policies: scalar, vector, and random. Each policy contributes 30 landscapes with different random seeds. All utilities are min-max normalized within each landscape to the range $[0, 1]$.

Generation policies. The scalar policy assigns each attack a latent scalar strength and aggregates the selected strengths. The vector policy assigns each attack a vector in a latent coverage space and scores the subset by the coverage induced by the selected vectors. The random policy starts from per-attack base potentials and adds deterministic subset-level variation. Across all policies, the latent score is further shaped by shared mechanisms that mimic patterns expected in real defense utilities: diminishing gains as subsets grow, a weak diversity bonus for covering more synthetic attack families, and pairwise interactions that can model either redundancy or complementarity between attacks.

Exact references. For each landscape, we enumerate all 2^{20} subset utilities and compute exact AttackSHAP values from the full table. These exact values serve as references for evaluating surrogate utility learning, full-vector AttackSHAP estimation, top-k attribution recovery, and budgeted subset selection. For surrogate learning, sampled subset utilities are split into training and validation subsets, and held-out utility prediction is evaluated on unqueried subsets from the same landscape.

Utility-landscape visualization. Figure 4 visualizes one representative synthetic landscape. It shows how subset utility varies across subset sizes and illustrates that neighboring subsets can have different marginal gains even when their sizes are similar. This provides an intuitive view of why subset-level utility cannot always be recovered by ranking attacks independently.

C Additional Synthetic Subset-Selection Results

The main text reports Normalized Gain for synthetic subset selection because it normalizes each utility landscape by its own random-baseline and oracle gap before averaging. For completeness, Table 6 reports the corresponding raw subset utility $v_\theta(S_k(m))$, averaged over the same 90 minmax utility landscapes used in Table 4. Because Normalized Gain is averaged per landscape, it is not obtained by normalizing the marginal means in Table 6. The raw utilities show the same qualitative pattern as Normalized Gain: direct subset optimization improves over attribution-only top-k selection, while $\textsc{OracleSearch}_v$ remains the upper bound.

D Real LLM Safety Settings

We evaluate all $2^{10} = 1,024$ defense subsets in each setting: Qwen2.5-7B + SFT, Ministral-3-8B + ICD, and Ministral-3-8B + SFT. Qwen + SFT is the main-text setting; both Ministral settings provide additional comparisons.

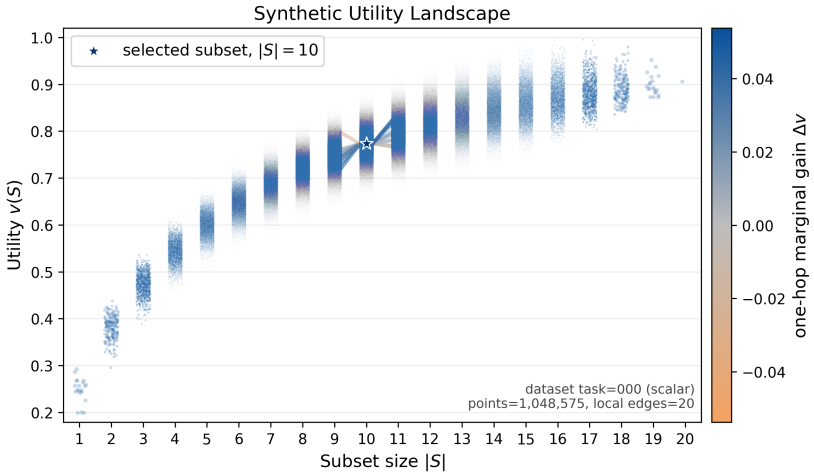


Figure 4: Example synthetic utility landscape. Each point corresponds to one attack subset, grouped by subset size $|S|$ on the x-axis and plotted by utility $v(S)$ on the y-axis. The highlighted star marks one selected subset, and local edges show one-hop neighboring subsets obtained by adding or removing one attack. Edge color indicates the one-hop marginal utility change Δv.

Table 6: Raw subset utility averaged over the same 90 synthetic utility landscapes. For each method m, $S_k(m)$ denotes the selected size-k subset, and each entry reports $v_\theta(S_k(m))$. These marginal means are reported for interpretability, and the Normalized Gain in Table 4 is computed per landscape before averaging.

Method	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$
Natural Baseline					
RANDOM	0.2941	0.4527	0.5479	0.6171	0.6663
Attribution Baselines					
TOPK-AOO$_v$	0.4456	0.5580	0.6471	0.6980	0.7329
TOPK-LOO$_v$	0.3497	0.5208	0.6344	0.7344	0.8055
TOPK-ϕ_v	0.4018	0.5641	0.6814	0.7592	0.8158
TOPK-$\hat{\phi}$	0.3828	0.5577	0.6682	0.7506	0.8203
A-MESS Solvers					
A-MESS-GREEDY	0.4456	0.6119	0.7143	0.7897	0.8449
A-MESS-SURROGATE	0.3791	0.5833	0.6958	0.7841	0.8527
Oracle Upper Bound					
ORACLESEARCH$_v$	0.4456	0.6365	0.7450	0.8253	0.8851

Defense construction. The defense behaviors are the first three JBB indices after shuffling with seed 2026. SFT uses three attacked requests per selected attack, each paired with a fixed refusal completion; a size-k subset therefore supplies $3k$ examples. Each subset starts from the base model and trains a fresh LoRA adapter for one epoch with batch size 1. LoRA uses rank 8, scaling 16, zero dropout, and text attention/MLP projections. AdamW uses initial learning rate 10^{-4} with cosine decay and zero weight decay; loss covers assistant-completion tokens only. ICD uses two fixed examples per selected attack, a safety instruction, and the current test request; the empty subset uses the original prompt. Native Ministral chat-template token IDs are retained for loading and completion masking.

Evaluation and overlap. Each defended model is evaluated on all 100 JBB and 440 SorryBench behaviors for all ten attacks, 100 safe XSTest requests, and tinyMMLU-100. The JBB evaluation includes the defense behaviors, so this full-benchmark evaluation is not strictly disjoint from defense construction. Target inference is greedy using vLLM; SFT uses Transformers and PEFT. General-capability responses have an 8-token cap; other original evaluation responses have a 256-token cap. The SorryBench Optimized Suffix supplement uses 512 tokens and the same frozen defended models. Its suffix assignments are fixed per question and target model, selected on SorryBench; the JBB suffix

Table 7: Attack categories used in the real LLM experiments.

Attack name	Brief description
Base64 Chu et al. (2025b)	Encodes the harmful request and asks the model to decode or operate on the encoded content.
Roleplay Shen et al. (2024)	Frames the request through a persona, fictional role, or simulated scenario to weaken refusal behavior.
Prefix Injection Wei et al. (2023a)	Prepends instructions that attempt to override the model's safety policy or system behavior.
Refusal Suppression Wei et al. (2023a)	Explicitly asks the model not to refuse, apologize, moralize, or mention safety constraints.
JSON Format Yoosuf et al. (2025)	Requests the answer in a structured format, such as JSON fields, to reduce natural-language refusal cues.
Affirmative Prefix Zhu et al. (2024a)	Starts the assistant response with an affirmative phrase so generation continues in a compliant direction.
Past Tense Andriushchenko & Flammarion (2025)	Rewrites the harmful instruction as a past-tense or historical description.
Low-resource Language Huang et al. (2026)	Translates or obfuscates the request through a lower-resource language or multilingual form.
Style Transfer Luo et al. (2026)	Asks the model to express the harmful content in a specific style, genre, or transformed surface form.
Optimized Suffix Zou et al. (2023); Zhou et al. (2025b); Zhu et al. (2024b)	Appends an optimized suffix or model-specific continuation pattern to induce harmful compliance.

remains a frozen transfer attack. This attack-construction and generation-budget difference is part of the evaluation protocol, not evidence of held-out attack generalization. PAIR is not included in these tables.

Scoring. Gemma-3-12B-it judges attack success and safe-request refusal; rates exclude invalid judgments and retain attempted/valid/invalid counts in the raw reports. TinyMMLU uses exact option parsing without a judge; refusals and unparseable answers count as incorrect out of 100, without retries. All analyses use the utility defined in Eq. (10), with component weights 0.9/0.05/0.05. For each attack, benchmark ASRs are averaged equally before calculating clipped relative risk reduction; all ten attacks contribute equally to $s_{\mathrm{ASR}}(S)$. Thus high utility describes the chosen safety-heavy tradeoff, not uniformly high capability or low absolute residual ASR.

E Attack Taxonomy

Table 7 lists the ten attack categories used across the real LLM experiments.

F Baseline and Solver Details

Attribution baselines. ASR ranks attacks by no-defense attack success rate and does not query the defender-side utility function. AOO measures empty-context marginal gain,

$$\mathrm{AOO}_i = v_\theta(\{a_i\}) - v_\theta(\emptyset), \tag{11}$$

while LOO measures full-context marginal loss,

$$\mathrm{LOO}_i = v_\theta(\mathcal{A}) - v_\theta(\mathcal{A} \setminus \{a_i\}). \tag{12}$$

AttackSHAP averages marginal utility over all subset contexts. In synthetic experiments, reference AttackSHAP is computed exactly from the full utility table. In surrogate-based estimation, AttackSHAP is computed by replacing true utility queries with $\hat{v}_\theta$.

Subset-selection baselines. For budget k, RANDOM averages over random size-k subsets. TOPK-ASR, TOPK-AOO, and TOPK-LOO rank individual attacks by the corresponding score and return the top k. A-MESS-GREEDY adaptively queries true utilities and adds the attack with the largest observed marginal gain at each step. A-MESS-SURROGATE first learns $\hat{v}_\theta$ from sampled subset utilities and then searches the learned utility landscape. A true-AttackSHAP top-k baseline is reported only for synthetic experiments. When the full synthetic landscape is available, ORACLESEARCH exhaustively searches all size-k subsets and is used only as an upper bound.

Original synthetic surrogate training. In the original synthetic estimation and selection experiments, A-MESS-Surrogate is trained under utility-query budgets $M \in \{100, 300, 500, 700, 900\}$. For each task and budget, models are selected by validation error, and the selected surrogate is used for downstream AttackSHAP estimation and subset search. Because the synthetic utility table is fully enumerated, held-out evaluation can be performed on subsets that were never used for surrogate training.

G ADDITIONAL SINGLETON RANKING COMPARISONS

Figures 5 and 6 show all ten attacks for the two Ministral settings, using the same JBB-only ranking and relative-risk utility as the main text. Ministral + ICD has Spearman $\rho = 0.236$ and Kendall $\tau_b = 0.156$; Ministral + SFT has $\rho = 0.091$ and $\tau_b = 0.067$. These descriptive correlations come from one greedy evaluation per defense subset; they do not establish repeated-generation statistical significance.

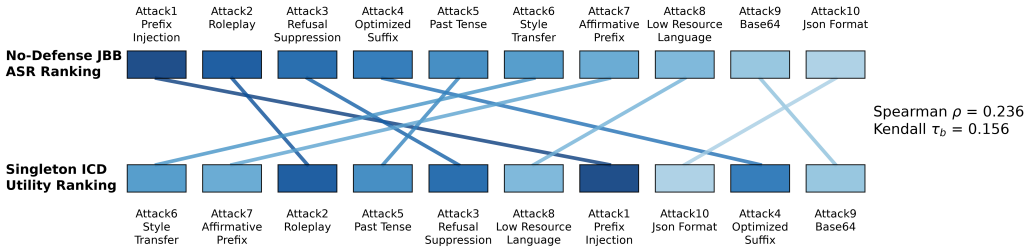


Figure 5: Attack success and defense utility yield different rankings for Ministral-3-8B + ICD. The upper row ranks attacks by no-defense JBB ASR; the lower row ranks their utility as individual sources of defense demonstrations. Lines connect the same attack across rankings.

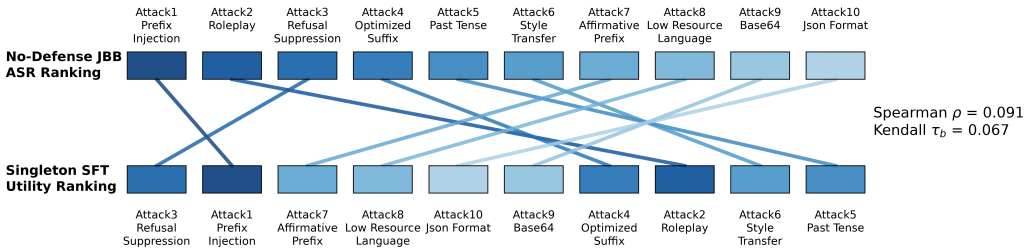


Figure 6: Attack success and defense utility yield different rankings for Ministral-3-8B + SFT. The upper row ranks attacks by no-defense JBB ASR; the lower row ranks their utility as individual sources of safety-training data. Lines connect the same attack across rankings.

H Additional Surrogate Prediction Results

Both settings use the same fixed 500 masks, train/test prefixes, architecture, and five model seeds as Table 3. The standard deviation measures training randomness on a fixed split. Ministral + ICD retains high rank correlations but its R^2 does not improve monotonically across the changing test sets. Ministral + SFT improves from $R^2 = 0.421$ to 0.933 between 100 and 400 training labels.

Table 8: Ministral-3-8B + ICD utility prediction on a fixed 500-subset pool. Mean ± sample std over model seeds 2026–2030; each row uses one fixed train/test split.

n_{train}:n_{test}	MAE ↓	R^2 ↑	Pearson ↑	Spearman ↑
100:400	0.015± 0.005	0.719± 0.197	0.919± 0.025	0.922± 0.021
200:300	0.011± 0.002	0.856± 0.039	0.948± 0.012	0.948± 0.010
300:200	0.012± 0.005	0.831± 0.128	0.958± 0.007	0.958± 0.007
400:100	0.013± 0.007	0.804± 0.191	0.963± 0.005	0.963± 0.006

Table 9: Ministral-3-8B + SFT utility prediction on a fixed 500-subset pool. Mean ± sample std over model seeds 2026–2030; each row uses one fixed train/test split.

n_{train}:n_{test}	MAE ↓	R^2 ↑	Pearson ↑	Spearman ↑
100:400	0.062± 0.005	0.420± 0.167	0.684± 0.079	0.449± 0.047
200:300	0.042± 0.005	0.740± 0.087	0.864± 0.048	0.638± 0.029
300:200	0.032± 0.002	0.889± 0.009	0.950± 0.003	0.722± 0.035
400:100	0.027± 0.004	0.933± 0.015	0.970± 0.008	0.785± 0.048

I Real Subset Selection: Acquisition and Additional Results

Table 5 and Tables 10–11 use one acquisition trajectory per setting, with seed fixed as 2026. The initial 100 subsets are sampled using a uniform-cardinality warm start. Subsequent batches contain 25 subsets drawn from the full unqueried coalition universe. Five guide MLPs ($J = 5$, seeds 2026–2030) estimate prediction variance, and acquisition balances queried cardinalities before maximizing variance within the chosen layer. Ties use ascending coalition-mask order. Only queried labels are available to acquisition. At 500 labels, a separate final MLP predicts all 1,024 coalition utilities; selection maximizes predicted utility among subsets of size k. The returned subset's cached true utility is reported, without replacing predictions by known labels during selection. Guide and final models use the same fixed architecture, full-batch Adam at learning rate 0.005, and 500 epochs as Table 3. There is no validation search, early stopping, or refit.

Random samples 20 distinct subsets per cardinality using seed 2026 and the fixed combinations order, shared across settings. AOO uses singleton-minus add-one-on utility; LOO uses full-minus leave-one-out utility; greedy repeatedly maximizes true marginal utility. TopK-ASR ranks equal-weighted no-defense JBB and SorryBench ASR, independently of the JBB-only display IDs in the ranking figures.

J Utility-Weight Sensitivity in Qwen2.5-7B + SFT

Tables 12 and 13 repeat the Qwen2.5-7B + SFT subset-selection comparison with weights $(0.8, 0.1, 0.1)$ and $(0.7, 0.15, 0.15)$ on $(s_{\text{ASR}}, s_{\text{ORR}}, \text{Acc})$, instead of the main-text $(0.9, 0.05, 0.05)$. The underlying response evaluations are unchanged; coalition utilities, acquisition trajectories, surrogate fits, and utility-based baseline selections are recomputed for each definition. Each surrogate uses 500 acquired labels and the seed-2027 final fit; Random averages the same 20 sampled subsets per size. For both alternatives, the surrogate reaches the exhaustive optimum at $k = 4$ and comes within 0.0054 of it at $k = 2$. Raw utility values should not be compared across weight definitions.

Table 10: Ministral-3-8B + ICD defense utility under attack-subset selection with $k = 2$ and $k = 4$. We compare the selected subsets, their observed utility, and the gain over the no-defense baseline (ΔU). Bold values indicate the best utility at each subset size.

Method	Selected subset	Utility↑	ΔU↑	Method	Selected subset	Utility↑	ΔU↑
NO DEFENSE	–	0.0835	0.0000	NO DEFENSE	–	0.0835	0.0000
Baselines with budget $k = 2$				*Baselines with budget $k = 4$*			
RANDOM	20 random subsets	0.8115	0.7280	RANDOM	20 random subsets	0.8076	0.7241
TOPK-ASR	Attack_1, Attack_2	0.8538	0.7703	TOPK-ASR	Attack_1, Attack_2, Attack_3, Attack_4	0.8600	0.7765
TOPK-AOO	Attack_6, Attack_7	0.8272	0.7437	TOPK-AOO	Attack_2, Attack_5, Attack_6, Attack_7	0.8863	0.8028
TOPK-LOO	Attack_2, Attack_3	0.8370	0.7535	TOPK-LOO	Attack_2, Attack_3, Attack_5, Attack_6	0.8924	0.8089
A-MESS Solvers with budget $k = 2$				*A-MESS Solvers with budget $k = 4$*			
A-MESS-GREEDY	Attack_2, Attack_6	**0.8569**	**0.7734**	A-MESS-GREEDY	Attack_2, Attack_3, Attack_5, Attack_6	0.8924	0.8089
A-MESS-SURROGATE	Attack_2, Attack_6	**0.8569**	**0.7734**	A-MESS-SURROGATE	Attack_2, Attack_3, Attack_6, Attack_7	**0.9035**	**0.8200**

Table 11: Ministral-3-8B + SFT defense utility under attack-subset selection with $k = 2$ and $k = 4$. We compare the selected subsets, their observed utility, and the gain over the no-defense baseline (ΔU). Bold values indicate the best utility at each subset size.

Method	Selected subset	Utility↑	ΔU↑	Method	Selected subset	Utility↑	ΔU↑
NO DEFENSE	–	0.0835	0.0000	NO DEFENSE	–	0.0835	0.0000
Baselines with budget $k = 2$				*Baselines with budget $k = 4$*			
RANDOM	20 random subsets	0.4042	0.3207	RANDOM	20 random subsets	0.8823	0.7988
TOPK-ASR	Attack_1, Attack_2	0.5181	0.4346	TOPK-ASR	Attack_1, Attack_2, Attack_3, Attack_4	0.8599	0.7764
TOPK-AOO	Attack_1, Attack_3	0.5090	0.4255	TOPK-AOO	Attack_1, Attack_3, Attack_7, Attack_8	0.8832	0.7997
TOPK-LOO	Attack_3, Attack_{10}	0.4225	0.3390	TOPK-LOO	Attack_3, Attack_5, Attack_7, Attack_{10}	0.8545	0.7710
A-MESS Solvers with budget $k = 2$				*A-MESS Solvers with budget $k = 4$*			
A-MESS-GREEDY	Attack_3, Attack_5	0.5365	0.4530	A-MESS-GREEDY	Attack_2, Attack_3, Attack_5, Attack_9	0.9030	0.8195
A-MESS-SURROGATE	Attack_1, Attack_5	**0.5855**	**0.5020**	A-MESS-SURROGATE	Attack_1, Attack_6, Attack_8, Attack_9	**0.9393**	**0.8558**

K COALITION UTILITY ACQUISITION PROTOCOL

We compare UNIFORM-COALITION, UNIFORM-SIZE, and A-MESS-ACQUISITION on 90 synthetic landscapes, comprising 30 scalar, 30 vector, and 30 random landscapes. Each landscape has $n = 20$ and utilities in $[0, 1]$; these previously studied landscapes are not an untouched test set. We report query budgets from 100 to 1,000 in increments of 25, counting distinct queried coalitions. Acquisition seed 2026 controls sampling and cardinality tie-breaking. UNIFORM-COALITION samples without replacement from all coalitions, whereas UNIFORM-SIZE balances cardinalities and samples uniformly without replacement within each non-exhausted size. A-MESS-ACQUISITION shares its first 100 queried coalitions with UNIFORM-SIZE, then acquires batches of 25 using Algorithm 1.

At each round, five guide MLPs are trained from scratch on the same observed pairs using seeds 2026–2030, without bootstrap resampling. The candidate set contains *every unqueried coalition* among the 2^{20} possible coalitions; no candidate subsampling is used. All candidates are scored in evaluation mode, with predictions computed in chunks solely to limit memory use. The disagreement score in Eq. 3 is

$$d(S) = \frac{1}{J}\sum_{j=1}^{J}\big(f_j(S) - \bar{f}(S)\big)^2, \qquad \bar{f}(S) = \frac{1}{J}\sum_{j=1}^{J} f_j(S), \qquad J = 5. \tag{13}$$

Selection first chooses a least queried available size, then maximizes disagreement within that size. A seeded per-round priority breaks size ties, and ascending coalition-mask order breaks score ties. Counts and candidate availability update after each choice; guides and scores remain fixed within the batch. The selected utilities are queried before the next round begins.

All guides and final predictors use a 20–128–128–64–1 MLP with ReLU and dropout 0.1 after the first two hidden layers. Training uses full-batch Adam, learning rate 0.005, and 500 epochs of MSE minimization, without validation-based selection or early stopping. At each reported budget, five separate final MLPs (seeds 2026–2030) are fitted to the same acquisition prefix and evaluated individually. The committee guides acquisition only; final predictions are not ensembled. Empty-coalition predictions are fixed to zero for evaluation. Unqueried utilities and exact Shapley values are used only for evaluation, never for acquisition.

Table 12: Qwen2.5-7B + SFT subset selection with utility weights $(0.8, 0.1, 0.1)$.

Method	Selected subset	Utility↑	ΔU↑	Method	Selected subset	Utility↑	ΔU↑
NO DEFENSE	–	0.1670	0.0000	NO DEFENSE	–	0.1670	0.0000
Baselines with budget $k = 2$				*Baselines with budget $k = 4$*			
RANDOM	20 random subsets	0.3328	0.1658	RANDOM	20 random subsets	0.6344	0.4674
TOPK-ASR	Attack_1, Attack_6	0.3269	0.1599	TOPK-ASR	Attack_1, Attack_2, Attack_3, Attack_6	0.6477	0.4807
TOPK-AOO	Attack_1, Attack_3	**0.4371**	**0.2701**	TOPK-AOO	Attack_1, Attack_3, Attack_5, Attack_{10}	0.6788	0.5118
TOPK-LOO	Attack_7, Attack_9	0.2486	0.0816	TOPK-LOO	Attack_2, Attack_7, Attack_8, Attack_9	0.5232	0.3562
A-MESS Solvers with budget $k = 2$				*A-MESS Solvers with budget $k = 4$*			
A-MESS-GREEDY	Attack_1, Attack_3	**0.4371**	**0.2701**	A-MESS-GREEDY	Attack_1, Attack_3, Attack_4, Attack_5	0.7290	0.5620
A-MESS-SURROGATE	Attack_3, Attack_5	0.4317	0.2647	A-MESS-SURROGATE	Attack_3, Attack_4, Attack_5, Attack_9	**0.8460**	**0.6790**

Table 13: Qwen2.5-7B + SFT subset selection with utility weights $(0.7, 0.15, 0.15)$.

Method	Selected subset	Utility↑	ΔU↑	Method	Selected subset	Utility↑	ΔU↑
NO DEFENSE	–	0.2505	0.0000	NO DEFENSE	–	0.2505	0.0000
Baselines with budget $k = 2$				*Baselines with budget $k = 4$*			
RANDOM	20 random subsets	0.3945	0.1440	RANDOM	20 random subsets	0.6576	0.4071
TOPK-ASR	Attack_1, Attack_6	0.3898	0.1393	TOPK-ASR	Attack_1, Attack_2, Attack_3, Attack_6	0.6699	0.4194
TOPK-AOO	Attack_1, Attack_3	**0.4862**	**0.2357**	TOPK-AOO	Attack_1, Attack_3, Attack_5, Attack_{10}	0.6971	0.4466
TOPK-LOO	Attack_7, Attack_9	0.3213	0.0708	TOPK-LOO	Attack_2, Attack_7, Attack_8, Attack_9	0.5622	0.3117
A-MESS Solvers with budget $k = 2$				*A-MESS Solvers with budget $k = 4$*			
A-MESS-GREEDY	Attack_1, Attack_3	**0.4862**	**0.2357**	A-MESS-GREEDY	Attack_1, Attack_3, Attack_4, Attack_5	0.7410	0.4905
A-MESS-SURROGATE	Attack_3, Attack_5	0.4809	0.2304	A-MESS-SURROGATE	Attack_3, Attack_4, Attack_5, Attack_9	**0.8377**	**0.5872**

We report raw Shapley MSE, $n^{-1}\sum_i(\hat{\phi}_i - \phi_i)^2$, and subset selection gap. For $\hat{S}_k \in \arg\max_{|S|=k} \hat{v}(S)$, the latter is

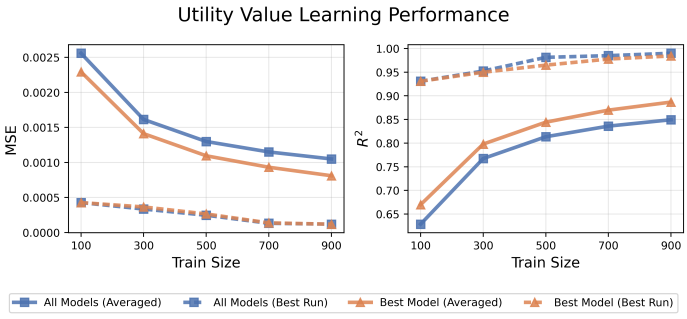
$$\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\left[\max_{|S|=k} v(S) - v(\hat{S}_k)\right], \qquad \mathcal{K} = \{1, 3, 5, 7, 9\}. \tag{14}$$

Neither metric is range-normalized. We average the five model seeds within each landscape, then average across landscapes. Shading shows pointwise 95% percentile confidence intervals from 5,000 bootstrap resamples of whole landscapes within each family (seed 20260920). The same resamples are shared across methods and budgets; intervals condition on the five fixed model seeds. Over the displayed budget range, A-MESS-ACQUISITION reduces Shapley error more rapidly in the early acquisition stages and reaches lower subset selection gaps than passive acquisition.

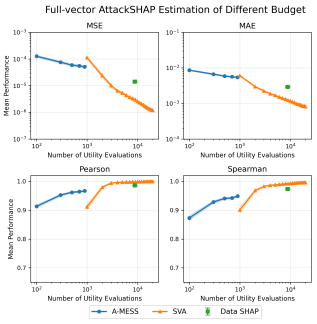
L ADDITIONAL UTILITY LEARNING AND ATTRIBUTION DIAGNOSTICS

We evaluate Qwen2.5-7B + SFT utility prediction on a fixed pool of 500 subsets. We shuffle all 1,024 coalition masks with seed 2026 and retain the first 500 in that order. The first N subsets form the training set and the remaining $500 - N$ form the test set, for $N \in \{100, 200, 300, 400\}$. Each split is fixed across five model seeds (2026–2030), with no test labels used during training. The surrogate is a ReLU MLP with hidden widths 128–128–64 and dropout 0.1 after the first two hidden layers. We use full-batch Adam, learning rate 0.005, and 500 epochs, without a validation split or early stopping.

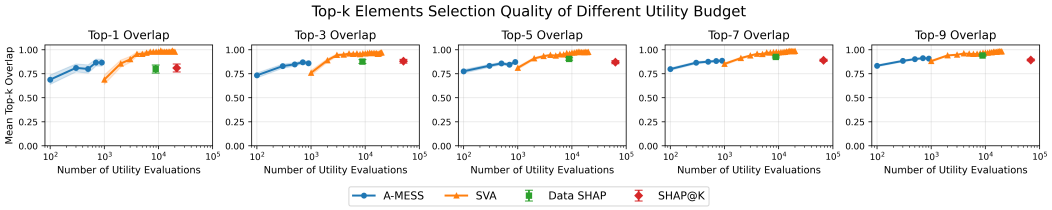
Table 3 reports means and sample standard deviations over model seeds, not over independently resampled splits. From 100 to 400 training subsets, MAE decreases from 0.065 to 0.037 and R^2 increases from 0.833 to 0.952. The training prefixes are nested, but test sets differ across rows; these are not repeated measurements on one fixed test set. Appendix Tables 8 and 9 report the corresponding results for both Ministral defenses.



(a) Surrogate utility learning performance. We evaluate how accurately the learned surrogate utility model approximates the ground-truth subset utility under different numbers of sampled utility evaluations. The surrogate model increasingly approximates the underlying subset utility landscape as more sampled subset evaluations are provided. R^2 achieves more than 0.9 for the best performing surrogate models.



(b) Full-vector AttackSHAP estimation. We compare AttackSHAP estimation quality under different utility budgets using error-based and correlation-based metrics.



(c) Top-k attack attribution quality. We evaluate whether estimated attribution scores recover the top-k reference attacks.

Figure 7: Synthetic evaluation of A-MESS-SURROGATE. Panel 7a evaluates surrogate utility prediction under increasing utility-query budgets. Panel 7b compares surrogate-based AttackSHAP estimates with reference full-vector AttackSHAP values. Panel 7c measures top-k overlap with the attacks identified by reference AttackSHAP.