Private Evaluation Protocol: Contamination-Resistant Black-Box Evaluation via Public-Slice Surrogates

Md. Mahadi Hassan
University of Central Florida
mdmahadi.hassan@ucf.edu

Sazzadur Rahaman
University of Arizona
saaz@arizona.edu

Yusuf Sarwar Uddin
University of Missouri–Kansas City
muddin@umkc.edu

Santu Karmaker
University of Central Florida
santu@ucf.edu

Abstract

Existing paradigms for black-box LLM evaluation by independent third parties share a structural weakness: the testing benchmark must reach the model being evaluated, enabling model developers to game future evaluations through logging or retraining. Ideally, rigorous evaluation would grant evaluators direct access to the model so testing can occur without exposing benchmark contents. However, proprietary black-box models cannot be released to external evaluators, creating a fundamental barrier to independent evaluation. To address this gap, we propose the Private Evaluation Protocol (PEP), a surrogate-based framework for independent evaluation of black-box models without sending the private benchmark to the model developer. PEP releases a small public slice of a held-out benchmark, queries the target model only on that slice, and trains a surrogate from the resulting responses, balancing fidelity to the target model against leakage of the private remainder over a tunable revelation budget. Across six benchmarks, composite leakage is strongly associated with behavioral exposure across the evaluated dataset–budget settings ($r = 0.788$), and fine-tuned surrogates reach 0.85–0.90 agreement on template-heavy datasets while plateauing below 0.73 on reasoning-heavy datasets. The optimal revelation budget is dataset-dependent, and PEP estimates black-box performance with 5.88 percentage-point MAE, showing the feasibility of private surrogate-based evaluation for structured tasks while protecting the privacy of the held-out benchmark.

1 Introduction

As LLMs are increasingly deployed in regulated domains, evaluation has become an important accountability mechanism. Regulators and auditors use benchmark-based assessments to verify whether these systems meet behavioral standards around bias, safety, and fairness (Raji et al., 2020). This setup creates a structural incentive for the party being evaluated to optimize their performance on the benchmark rather than the underlying capability. In practice, that incentive is amplified by Internet-scale training, which can expose models to benchmark instances, inflate scores, and blur the distinction between memorization and genuine generalization, making public benchmarks unreliable (Golchin and Surdeanu, 2024; Kandpal et al., 2024). Researchers have responded with overlap audits (Xu et al., 2024), membership inference (Tong et al., 2025), statistical tests (Dekoninck et al., 2024), and contamination-resistant benchmarking (Chen et al., 2025b). Unfortunately, they all require the benchmark to reach the model being evaluated. For independent evaluation (Raji et al., 2022; Mökander et al., 2024), this creates a two-sided dilemma. Sending the benchmark to the model risks retention through logging or retraining, while keeping it private would require releasing the model to the evaluator, which proprietary black-box systems do not permit. The evaluator, therefore, needs a protocol that eliminates benchmark exposure by design, rather than detecting and mitigating contamination after the fact.

Cryptographic approaches such as MPC (Sayyad, 2020; Knott et al., 2021), TEEs (Rajore et al., 2024), and FHE (Gouert et al., 2023) eliminate exposure without revealing inputs, but they require the model developer to participate in the secure protocol, which is incompatible with existing model vendors.

To address this gap, we propose the *Private Evaluation Protocol (PEP)*, a surrogate-based framework for independent evaluation of a black-box model without the private scoring split ever reaching the model developer. The evaluator reveals a small budgeted public slice of the benchmark and queries the target model only on that slice. The resulting input-output pairs provide behavioral supervision for constructing an evaluator-controlled

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 3726–3749
November 6-10, 2026 ©2026 Association for Computational Linguistics

surrogate, which is then scored on the private split entirely within the evaluator's infrastructure.

PEP does not require the surrogate to reconstruct the target model globally. It only requires the revealed responses to contain enough information for the surrogate to approximate the target model's behavior on the benchmark distribution well enough to estimate performance on the unrevealed private split. Here, our goal is to establish the feasibility of this surrogate-based evaluation paradigm under a controlled setting, rather than to present a production-ready certification protocol. We study PEP on multiple-choice question answering (MCQA), where target–surrogate behavioral agreement and private-set estimation error can be defined consistently across benchmarks. Extending PEP to open-ended generation, such as summarization (Salvador et al., 2025) or music generation (Kader et al., 2026), or executable tasks such as IT automation (Hassan et al., 2026), would require task-specific definitions of behavioral equivalence and corresponding evaluation metrics, which we leave to future work.

This design immediately raises a concern about structural leakage, i.e., releasing even a small portion of the benchmark may expose the private remainder through shared templates, lexical patterns, or semantic similarity. We therefore ask: **RQ1:** *Can structural leakage from the public slice to the private remainder be reliably quantified before any model is queried, and does it predict actual behavioral exposure?*

Assuming leakage is measurable, the protocol's usefulness depends entirely on how faithfully the surrogate reproduces the target model's behavior on held-out data it has never seen. A surrogate trained only on a small public slice may capture surface-level patterns on structured datasets but fail entirely on benchmarks that require multi-step reasoning, where the decision boundary cannot be recovered from input-output pairs alone. This raises a second question: **RQ2:** *Does a surrogate constructed from black-box responses on a small public slice faithfully reproduce the target model's behavior on the held-out split, and under what dataset conditions does this break down?*

Releasing more data improves surrogate quality but increases leakage; releasing less data preserves the benchmark but degrades the surrogate. Even if leakage is measurable and surrogates are individually viable, the protocol is only operable if a revelation budget exists where both conditions are simultaneously satisfiable. Whether such points also yield useful accuracy estimates for candidate models adds a further empirical question: **RQ3:** *Does PEP reveal favorable privacy–utility operating points, and do those points yield useful candidate accuracy estimates on the private split?*

If no such operating point exists in practice, the protocol is theoretically motivated but practically inoperable. We evaluate all three conditions across six MCQA benchmarks spanning bias, fairness, and reasoning, sweeping eight revelation budgets from 1% to 30% and instantiating two surrogate creation regimes: fine-tuned surrogates that adapt an open-source model via LoRA (Hu et al., 2021) on the public slice, and few-shot surrogates that steer behavior of open-source LLMs through in-context learning with no parameter updates. Our contributions are as follows.

1. **Private Evaluation Protocol.** We introduce PEP, a surrogate-based protocol to formalize independent benchmark evaluation as a revelation regime, operating through black-box API access with a tunable revelation budget and a composite leakage audit.
2. **Feasibility Evidence.** We evaluate all three conditions empirically. Leakage is measurable before querying ($r = 0.788$ against behavioral exposure), fine-tuned surrogates reach 0.85–0.90 fidelity on structured datasets at low budgets, and at the validation-selected revelation budget, PEP estimates accuracy with 5.88 percentage-point MAE. These results show that private surrogate-based evaluation can support practical black-box evaluation for structured evaluation tasks while preserving the held-out benchmark.

2 Related Work

The Contamination Problem. LLMs trained on Internet-scale corpora routinely memorize benchmark instances, making genuine generalization indistinguishable from test-set recall (Xu et al., 2024; Chen et al., 2025b; Sainz et al., 2023). The community has built a broad detection arsenal spanning membership inference (Duan et al., 2024; Mireshghallah et al., 2022), prompt-completion probes (Golchin and Surdeanu, 2024; Shi et al., 2024), and statistical performance comparisons (Dekoninck et al., 2024; Oren et al., 2023). Despite their diversity, detection approaches share the same weakness: exposure must occur before detection can run, and by then it is irreversible (Eriks-

son et al., 2025).

Contamination Mitigation Strategies. Benchmark redesign approaches prevent contamination by refreshing content through recent events, real user queries (White et al., 2025; Lin et al., 2025), generating controlled behavioral variation (Sun et al., 2025; Ma et al., 2021; Ribeiro et al., 2020), or aggregating across many datasets (Liang et al., 2023). These approaches reduce the risk of prior exposure but do not address the gaming scenario: the benchmark must still reach the model developer during evaluation, leaving the contamination pathway open. Cryptographic approaches such as TEEs, MPC, and FHE (Rajore et al., 2024; Sayyad, 2020; Knott et al., 2021; Gouert et al., 2023) eliminate this exposure entirely by enabling evaluation without revealing inputs. However, all cryptographic methods require the model developer to participate in the secure protocol, which is not practical.

Surrogate Construction and Black-Box Distillation. A white-box student trained on black-box teacher outputs can recover task-specific decision boundaries and reasoning chains through output supervision alone (Hinton et al., 2015; Kim and Rush, 2016; Chen et al., 2024; Magister et al., 2023), with recent work extending this to on-policy and reverse-KL objectives (Ye et al., 2026; Gu et al., 2024). In the parameter-free regime, in-context learning steers behavior entirely through exemplar selection (Brown et al., 2020), and coverage-maximizing, semantically similar, and diversity-oriented strategies consistently outperform random baselines on structured tasks (Min et al., 2022; Rubin et al., 2022; Zhang et al., 2022).

Positioning and Novelty. PEP sits at the intersection of these bodies of work without being reducible to any. Prior detection methods assume the benchmark has already reached the model; benchmark redesign methods assume the benchmark can be made public; cryptographic methods assume vendor cooperation. PEP assumes none of these. What PEP contributes is the empirical question of whether a surrogate-based revelation regime is viable at all: whether structural leakage can be measured before querying, whether a surrogate trained on a small public slice can faithfully reproduce target model behavior on held-out data, and whether a revelation budget exists where both are simultaneously satisfiable. These are not design claims but empirical conditions, and establishing whether they hold is the central contribution.

3 Private Evaluation Protocol (PEP)

3.1 Setup and Motivation

The Private Evaluation Protocol (PEP) addresses a fundamental challenge in independent model evaluation: given a target model $\mathcal{M}$ and a held-out benchmark $\mathcal{D}$, how can an evaluator assess $\mathcal{M}$ on $\mathcal{D}$, when $\mathcal{M}$ is accessible only through a black-box API and sending $\mathcal{D}$ risks contamination for future evaluations?

The protocol involves two parties. The *evaluator* maintains a private benchmark and seeks an independent assessment of $\mathcal{M}$ that does not rely on self-reported scores from the model developer. The *model developer* operates $\mathcal{M}$ behind a black-box API. Directly sending $\mathcal{D}$ to the model developer's API creates a contamination risk: queried items may be retained through standard logging and incorporated into future training, permanently invalidating $\mathcal{D}$ as an evaluation instrument. The evaluator, therefore, needs a protocol that produces a reliable assessment of $\mathcal{M}$ without $\mathcal{D}$ ever leaving the evaluator's infrastructure.

We draw inspiration from public-private key cryptography (Diffie and Hellman, 1976). Just as a public key allows interaction with a system without exposing the private key, we design a public slice $\mathcal{D}_{\text{pub}}$ that allows the model developer to demonstrate capabilities via API queries without exposing the held-out split $\mathcal{D}_{\text{priv}}$ used for actual scoring. Unlike a cryptographic key, $\mathcal{D}_{\text{pub}}$ is not formally isolated from $\mathcal{D}_{\text{priv}}$: structural similarity between the two splits may still allow inference about the private remainder. Whether this similarity can be bounded before any model is queried is an empirical question central to PEP's viability. PEP therefore provides benchmark separation by construction, rather than a formal cryptographic privacy guarantee.

3.2 Threat Model and Assumptions

Contamination Threat: PEP addresses a single core threat: benchmark contamination through model developer access. Any mechanism by which $\mathcal{D}_{\text{priv}}$ reaches the model developer, whether through passive API logging, deliberate data retention, or incorporation into future training, permanently invalidates the benchmark as an independent audit instrument. PEP's defense is structural: $\mathcal{D}_{\text{priv}}$ never leaves the evaluator's environment, eliminating the direct exposure pathway regardless of model developer intent.

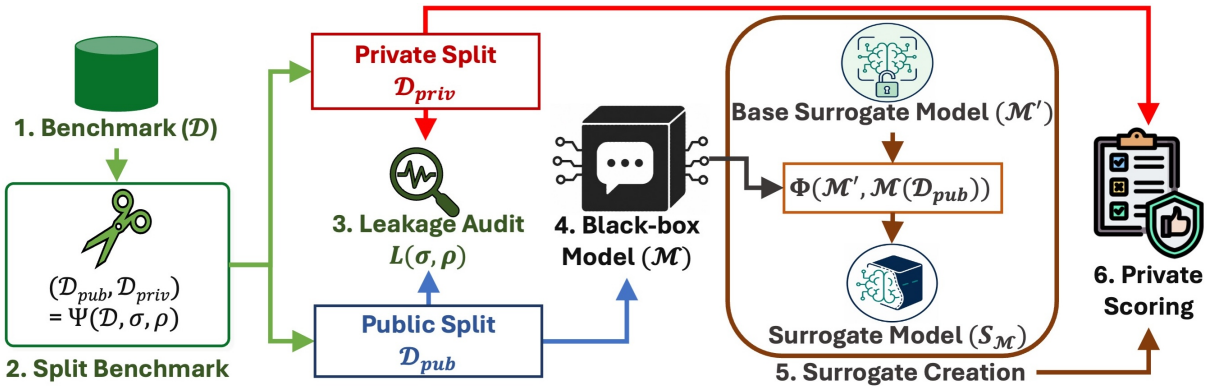


Figure 1: Deployment-time overview of PEP. The evaluator partitions $\mathcal{D}$ into a public interaction split $\mathcal{D}_{\text{pub}}$ and a private scoring split $\mathcal{D}_{\text{priv}}$. Only $\mathcal{D}_{\text{pub}}$ is submitted to the black-box model $\mathcal{M}$; $\mathcal{D}_{\text{priv}}$ remains local and is used only for leakage auditing and surrogate scoring.

Model Developer: We assume the model developer provides accurate responses when queried and that $\mathcal{M}$ resists membership inference attacks. We make no assumption about whether the developer would exploit retained data if given access to benchmark items. PEP therefore addresses benchmark exposure under standard honest-response black-box access; it is not a defense against a provider that detects evaluation queries and strategically alters its responses. Such adversarial response manipulation is outside the present threat model and remains future work.

Evaluator Capabilities: The evaluator has black-box API access to $\mathcal{M}$ and full white-box access to a set of open-source models $\mathcal{R} = \{R_1, \ldots, R_n\}$ any of which may be instantiated as a surrogate base model $\mathcal{M}'$. The evaluator initially controls the full dataset $\mathcal{D}$; after partitioning, only $\mathcal{D}_{\text{pub}}$ is released for interaction with $\mathcal{M}$ and used to create the surrogate, while $\mathcal{D}_{\text{priv}}$ remains entirely within the evaluator's trusted environment.

3.3 Protocol Overview

The core challenge of PEP is enabling independent evaluation of $\mathcal{M}$ on $\mathcal{D}_{\text{priv}}$ without ever sending $\mathcal{D}_{\text{priv}}$ to the developer. Since the evaluator cannot query $\mathcal{M}$ directly on $\mathcal{D}_{\text{priv}}$, the protocol must produce an estimate of $\mathcal{M}$'s behavior on the private split through a surrogate: a publicly runnable model trained to mimic $\mathcal{M}$'s behavior on D_{pub}.

The evaluator releases a small public slice $\mathcal{D}_{\text{pub}}$, queries $\mathcal{M}$ on this slice to obtain black-box responses, and uses those (question, response) pairs to construct a *surrogate*, $S_{\mathcal{M}}$ (a publicly runnable model that approximates $\mathcal{M}$'s behavior). The surrogate is built entirely from open-source models within the evaluator's infrastructure and can be evaluated freely on $\mathcal{D}_{\text{priv}}$ without involving $\mathcal{M}$. The quality of this evaluation pipeline depends directly on how faithfully $S_{\mathcal{M}}$ reproduces $\mathcal{M}$'s decision boundary on unseen private data. PEP does not require $S_{\mathcal{M}}$ to reconstruct $\mathcal{M}$'s global decision boundary; it requires sufficient behavioral agreement on the benchmark distribution for $S_{\mathcal{M}}$ to serve as a useful proxy on unseen private data.

This design requires three conditions to hold simultaneously: $S_{\mathcal{M}}$ must faithfully reproduce $\mathcal{M}$'s behavior on $\mathcal{D}_{\text{priv}}$; $\mathcal{D}_{\text{pub}}$ must not expose $\mathcal{D}_{\text{priv}}$ through semantic, lexical, or structural similarity; and both must be satisfiable at a revelation budget small enough to be practically useful. Whether these conditions can be simultaneously satisfied is not guaranteed by design; determining whether they can be satisfied simultaneously is the empirical question this work addresses.

3.4 Protocol Design

We separate the protocol itself from the metrics used to evaluate it. The protocol specification defines the runtime computation: $\mathcal{M}$ is queried only on $\mathcal{D}_{\text{pub}}$, while $\mathcal{D}_{\text{priv}}$ remains local and is used for leakage auditing and surrogate scoring. Figure 1 summarizes the deployment-time workflow. We also define a set of metrics to assess the feasibility of instantiating such a protocol.

3.4.1 Functional Specification

Dataset Partitioning. We introduce a revelation function Ψ that partitions $\mathcal{D}$ based on strategy σ and budget ρ:

$$(\mathcal{D}_{\text{pub}}, \mathcal{D}_{\text{priv}}) = \Psi(\mathcal{D}, \sigma, \rho) \tag{1}$$

where $\mathcal{D}_{\text{pub}} \cap \mathcal{D}_{\text{priv}} = \emptyset$ and $|\mathcal{D}_{\text{pub}}| = \rho \cdot |\mathcal{D}|$. The strategy σ specifies how items are selected for $\mathcal{D}_{\text{pub}}$, and the budget $\rho \in (0, 1)$ is the primary lever governing the privacy-utility tradeoff.

Leakage Measurement. We quantify information leakage from $\mathcal{D}_{\text{pub}}$ to $\mathcal{D}_{\text{priv}}$ via nearest-neighbor similarity:

$$L(\sigma, \rho) = \mathbb{E}_{x \sim \mathcal{D}_{\text{priv}}} \left[\max_{x' \in \mathcal{D}_{\text{pub}}} s(x, x') \right] \quad (2)$$

where $s(x, x')$ measures proximity between a private item and its nearest public neighbor. $L(\sigma, \rho)$ is an empirical measure of structural exposure between the public and private splits, rather than a formal privacy guarantee.

Surrogate Construction. Given black-box responses from $\mathcal{M}$ on $\mathcal{D}_{\text{pub}}$, we construct a publicly runnable surrogate:

$$S_{\mathcal{M}} = \Phi\big(\mathcal{M}',\ \{(x,\ \mathcal{M}(x))\}_{x \in \mathcal{D}_{\text{pub}}}\big) \quad (3)$$

where $\mathcal{M}'$ is an open-source model within the evaluator's infrastructure and Φ is the surrogate construction method, instantiated in Section 4. $\mathcal{M}$ is queried only on $\mathcal{D}_{\text{pub}}$; the surrogate $S_{\mathcal{M}}$ is thereafter used to estimate $\mathcal{M}$'s behavior on $\mathcal{D}_{\text{priv}}$ without any further interaction with $\mathcal{M}$.

PEP Runtime Estimate. The quantity the evaluator ultimately wants is the target model's private-set performance, $A_{\mathcal{M}}(\mathcal{D}_{\text{priv}})$. However, this quantity is not observable in deployment because computing it would require querying $\mathcal{M}$ on $\mathcal{D}_{\text{priv}}$. PEP therefore reports the surrogate's score on the private split:

$$\widehat{A}_{\text{PEP}}(\mathcal{M}; \mathcal{D}_{\text{priv}}) = A_{S_{\mathcal{M}}}(\mathcal{D}_{\text{priv}}). \quad (4)$$

This runtime output specified by PEP is computable as the evaluator holds $\mathcal{D}_{\text{priv}}$, controls $S_{\mathcal{M}}$, and scores the benchmark locally. Since $\mathcal{M}$ is never queried on $\mathcal{D}_{\text{priv}}$, the estimate's exact error is unobservable at deployment and should be interpreted alongside observable diagnostics such as leakage and surrogate stability. Like any validation-based protocol, deployment extrapolates from prior validation to new target models. PEP makes this uncertainty explicit: evaluators can validate on diverse reference candidates and similar benchmarks, then monitor leakage and inter-surrogate stability at runtime. These diagnostics do not certify the estimation gap, but they indicate when estimates should be trusted or treated conservatively.

Deployment procedure. In deployment, the evaluator first selects a conservative revelation configuration (σ, ρ) using prior validation on reference models together with the structural leakage observable for the benchmark. The evaluator then releases $\mathcal{D}_{\text{pub}}$, queries $\mathcal{M}$ only on this slice, constructs one or more surrogates from the returned responses, and scores them locally on $\mathcal{D}_{\text{priv}}$. Inter-surrogate stability provides an additional candidate-specific diagnostic after surrogate construction. If the observed diagnostics fall outside the range supported by prior validation, the resulting estimate should be interpreted conservatively or the evaluator may abstain from making a high-confidence assessment. These diagnostics provide evidence about when extrapolation is reasonable; they do not reveal or certify the true estimation error.

3.4.2 Evaluation Metrics

Any protocol that keeps a benchmark private must be evaluated on two axes: whether it limits exposure of the private split, and whether its reported estimate tracks the target model's true private-set behavior. The first axis is observable through leakage; the second requires controlled validation, because $\mathcal{M}$ is intentionally not queried on $\mathcal{D}_{\text{priv}}$ during deployment. The metrics below therefore evaluate the protocol under controlled validation; they are not runtime requirements.

Oracle Fidelity. The reliability of $\widehat{A}_{\text{PEP}}$ depends on whether the surrogate behaves like the target model on the private split. We define this latent quantity as oracle fidelity:

$$F^\star(\sigma, \rho) = \mathbb{E}_{x \sim \mathcal{D}_{\text{priv}}} \left[\mathbf{1}[S_{\mathcal{M}}(x) \approx \mathcal{M}(x)] \right], \quad (5)$$

where $\approx$ denotes task-appropriate behavioral equivalence. For tasks with non-discrete outputs, the indicator can be replaced by a task-specific agreement or similarity score. Oracle fidelity is not computable in deployment because it requires querying $\mathcal{M}$ on $\mathcal{D}_{\text{priv}}$. We use it only as a validation quantity in our experiments.

Validation Utility. To analyze whether PEP admits a favorable privacy–utility operating point, we define a validation utility that combines privacy (P) with oracle fidelity. Let $P(\sigma, \rho) = 1 - L(\sigma, \rho)$ and let $F_0^\star$ denote oracle fidelity before any revelation. We first compute the smoothed normalized gain:

$$G^\star_{\text{eff}}(\sigma, \rho) = \epsilon + (1-\epsilon) \max\left(0, \frac{F^\star(\sigma, \rho) - F_0^\star}{1 - F_0^\star}\right). \quad (6)$$

Surrogate quality is

$$S^\star(\sigma, \rho) = G^\star_{\text{eff}}(\sigma, \rho)^\eta \cdot F^\star(\sigma, \rho)^{1-\eta}. \quad (7)$$

The validation utility is then

$$U^\star(\sigma, \rho) = \frac{(1+\beta^2) S^\star(\sigma, \rho) P(\sigma, \rho)}{\beta^2 P(\sigma, \rho) + S^\star(\sigma, \rho)}. \quad (8)$$

Role	Model	Family
Candidate	`Llama-3.1-8B-Instruct`	Llama
Candidate	`Phi-3.5-mini-instruct`	Phi
Candidate	`aya-expanse-8b`	Aya
Surrogate	`Qwen2.5-14B-Instruct`	Qwen
Surrogate	`Gemma-3-12b-it`	Gemma
Surrogate	`Falcon3-7B-it`	Falcon

Table 1: Selected models. Surrogate and candidate families are non-overlapping by design.

$U^\star$ is used only to compare revelation budgets and protocol variants under controlled validation; it is not available for an unseen target model at deployment. Deployment therefore extrapolates from operating regions established on reference models, using benchmark-side leakage to guide revelation and candidate-specific inter-surrogate stability to assess the resulting estimate.

4 Experiment Design

We instantiate the abstract PEP framework from Section 3 using uniform random sampling as the revelation strategy σ. Uniform sampling makes no task-specific assumptions, allowing us to study the privacy–utility tradeoff without confounding it with optimized public-slice selection. We deliberately use this simple strategy to evaluate the feasibility of PEP under a conservative, task-agnostic instantiation; structured or adaptive revelation strategies may expose more candidate-specific behavioral information and are left for future work. We sweep $\rho \in \{0.01, 0.03, 0.05, 0.08, 0.10, 0.15, 0.20, 0.30\}$ across six benchmarks.

4.1 Models and Datasets

A core requirement of our experiments is minimizing the chance that candidate or surrogate models saw benchmark items during training. We therefore use benchmarks released strictly after January 2025 and pair them with open models whose publicly documented knowledge cutoffs predate that period. As far as public documentation allows, this provides a contamination-controlled validation setting for analyzing leakage, surrogate fidelity, and privacy-utility tradeoffs. We exclude *proprietary* models such as GPT (OpenAI et al., 2024), Claude (Anthropic, 2024), and Gemini (Comanici et al., 2025) from this controlled validation because their training cutoffs, update histories, and retention policies cannot be reliably verified, making contamination-controlled analysis under the same protocol impossible.

We evaluate PEP on six MCQA benchmarks across bias, fairness, and reasoning: *The Bias is in the Details (CognitiveBias)* (Knipper et al., 2025), *BharatBBQ* (Tomar et al., 2025), *PakBBQ* (Hashmat et al., 2025),*FairMedQA* (Xiao et al., 2026), *Logical-CSQA* (Junias and Pacheco, 2026), and *JustLogic* (Chen et al., 2025a). For comparability, we sample 3,000 instances per dataset when available and use a stratified 50% split as $\mathcal{D}_{\text{priv}}$, with $\mathcal{D}_{\text{pub}}$ drawn from the remaining pool at each revelation budget. We keep $\mathcal{D}_{\text{priv}}$ fixed across revelation budgets so that changes in leakage and surrogate fidelity reflect changes in ρ, rather than changes in the private evaluation set. We select candidate/surrogate models from non-overlapping families after self-consistency screening; see Appendix C.1 for more details.

We screened 14 open-source LLMs, all with training data cutoffs before January 1, 2025 (full list in Appendix C.2), retaining only models achieving at least 80% response consistency under majority vote across six prompt variations (TELeR taxonomy (Santu and Feng, 2023)) and six temperatures (0.01, 0.05, 0.10, 0.20, 0.50, 0.80) (details in Appendix C.3). From the self-consistent models, we select candidates and surrogates from strictly non-overlapping model families (Table 1), reducing the possibility that observed fidelity gains are driven primarily by shared model-family characteristics rather than the revealed target responses.

4.2 Protocol Instantiation

We instantiate Φ in two regimes. In the *fine-tuning* regime, $\mathcal{M}'$ is adapted with LoRA on public-slice input-output pairs $(x, \mathcal{M}(x))$, directly specializing the surrogate to reproduce the target's responses. In the *few-shot* regime, no parameters are updated; instead, selected public examples are prepended as demonstrations. We use $k = 8$ shots and compare random, diversity-based, disagreement-based, and uncertainty-based selection strategies. Full implementation details are in Appendix D.

4.3 Metric Instantiation

Fidelity. We instantiate behavioral equivalence from Eq. 5 by ranking free-form model responses against answer options using Sentence-BERT (Reimers and Gurevych, 2019) cosine similarity and measuring agreement on the top-ranked option. This fidelity instantiation is specific to multiple-choice QA and should be interpreted as a

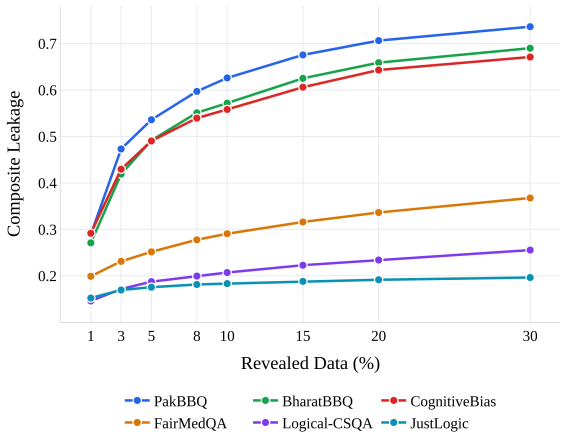


Figure 2: Composite leakage score across six datasets.

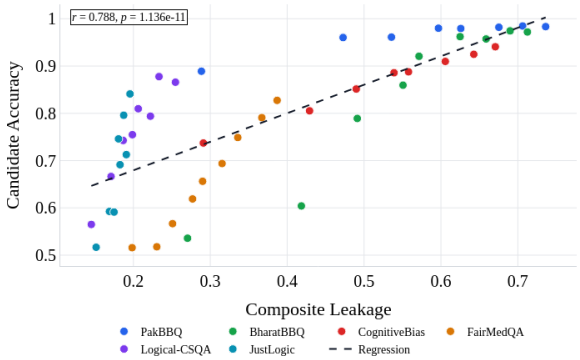


Figure 3: Static composite leakage predicts candidate accuracy gain on the private split.

task-specific proxy for behavioral agreement rather than a general evaluation recipe for open-ended generation settings. For structured outputs, such as the explicit belief representations evaluated by OmniToM (Bawatneh et al., 2026), agreement would need to account for the predicted structure rather than a single answer choice.

Leakage. We instantiate $s(x, x')$ from Eq. 2 using three static similarity signals between $\mathcal{D}_{\text{pub}}$ and $\mathcal{D}_{\text{priv}}$: token 5-gram containment, MinHash Jaccard similarity, and sentence-embedding cosine similarity. These signals capture direct phrase reuse, approximate lexical overlap, and paraphrase-level similarity. Together, they guard against any single metric failing on a dataset type.

Utility. We compute surrogate quality using Eq. 7 with $\eta = 0.25$ and smoothed normalized gain from Eq. 6, where F_0 is the average zero-shot fidelity across all candidates before any revelation. We set $\eta = 0.25$ using the Cobb-Douglas form and floor $\epsilon = 0.1$ following label-smoothing convention (Szegedy et al., 2016); $\beta = 2$ (Rijsbergen, 1979) places greater weight on privacy than surrogate quality, reflecting the regulated-domain use case where benchmark integrity is a hard constraint. As defined in Section 3.4.2, this utility is used only for controlled validation and is not assumed to be observable for an unseen deployment target. A sensitivity analysis of η and β is provided in Appendix A, confirming qualitative findings are stable across parameter choices.

5 Results

5.1 RQ1: Leakage Signal Validation

For PEP to be operationally trustworthy, the leakage metric must reflect genuine information transfer rather than superficial textual similarity, giving evaluators a reliable estimate of integrity cost before committing to a release.

Figure 2 shows the composite leakage score across all eight revelation budgets. A clear two-group structure emerges from the first budget point: PakBBQ, BharatBBQ, and CognitiveBias reach 0.67–0.73 by 30% revelation, while FairMedQA, Logical-CSQA, and JustLogic remain below 0.37. This split reflects a fixed property of benchmark structure: template-heavy benchmarks share dense surface patterns between splits, while reasoning benchmarks exhibit little structural overlap regardless of revelation budget.

To validate that these scores reflect genuine behavioral exposure, we fine-tune each candidate on $\mathcal{D}pub$ at every budget and measure accuracy gain on $\mathcal{D}priv$. Figure 3 shows the correlation with $r = 0.788$, $p = 1.136 \times 10^{-11}$ across 48 dataset-budget observations, with higher composite leakage consistently predicting larger accuracy gains.

Finding: Structural leakage is reliably measurable before any model is queried, and composite leakage scores serve as a design-time instrument for estimating benchmark exposure cost. Beyond protocol operation, the composite leakage score also serves as a pre-release audit instrument: benchmark designers can compute it at design time to identify and reduce inter-split structural overlap before publication.

5.2 RQ2: Surrogate Fidelity

Fig. 4 shows fine-tuned surrogate fidelity on $\mathcal{D}_{priv}$ across revelation budgets. The same two-group structure governs fidelity: PakBBQ, BharatBBQ, and CognitiveBias form an upper band starting above 0.77 at 1% and saturating near 0.85–0.90 by 10–15%, while FairMedQA, JustLogic, and

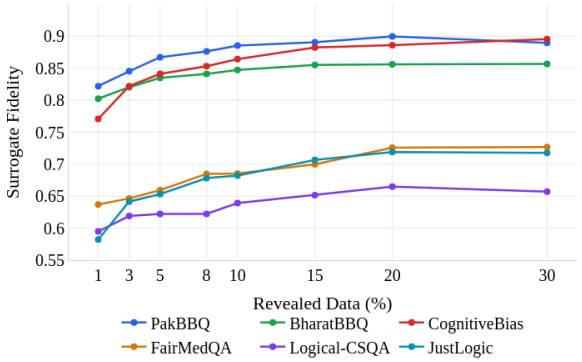


Figure 4: FT surrogate fidelity on $\mathcal{D}_{\text{priv}}$ vs. rev. budget.

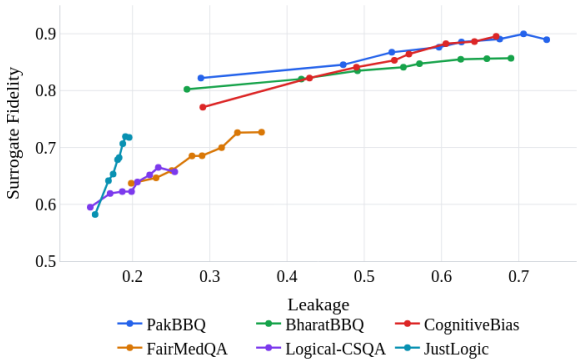


Figure 5: Leakage-fidelity frontier per dataset.

Logical-CSQA form a lower band that begins at 0.59–0.64 and plateaus below 0.73 even at 30% revelation. The early plateau on reasoning-heavy datasets reflects limited recoverable signal from black-box input-output pairs, though we cannot rule out surrogate capacity as a contributing factor.

Fig. 5 plots each dataset's trajectory in leakage-fidelity space as ρ increases. Upper-band datasets follow right-dominant paths where leakage grows faster than fidelity; lower-band datasets follow up-dominant paths, gaining fidelity at minimal leakage cost. This trajectory classification directly governs where the utility function peaks for each dataset.

Finding: Surrogate fidelity is strong on structured datasets in our setting, but it plateaus substantially lower on reasoning-heavy benchmarks, defining an important empirical boundary of PEP under the studied surrogate families and budgets.

5.3 RQ3: Protocol Utility and Validation

Fig. 6 shows the validation utility across all eight revelation budgets. Every dataset exhibits a clear peak, confirming that favorable privacy–fidelity tradeoffs exist. The peak is dataset-dependent: template-heavy datasets peak early because leakage grows faster than fidelity, while reasoning datasets peak later and decline more gradually.

To assess whether validation-selected budgets

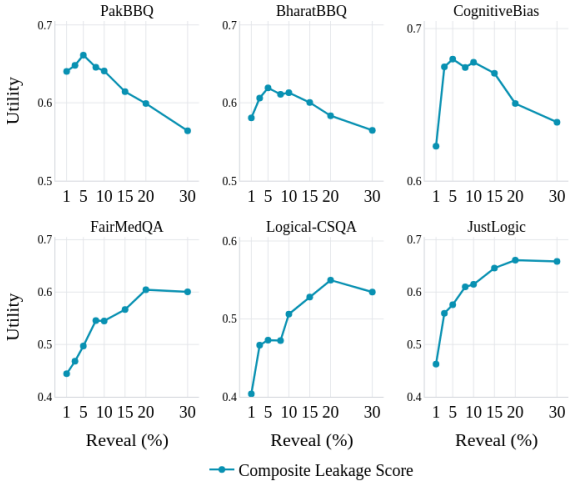


Figure 6: Protocol utility $U^*(\rho)$ per dataset.

Dataset	Aya	Llama	Phi	MAE
BharatBBQ	−3.86 26.19	−3.25 26.99	−4.68 26.19	3.93
CognitiveBias	**−8.84** 48.70	−1.32 54.22	**+8.03** 41.98	6.07
FairMedQA	**+13.20** 36.87	+1.15 57.80	+6.07 43.20	6.81
JustLogic	+6.93 45.84	+3.27 44.17	+1.48 46.04	3.89
Logical-CSQA	+5.98 45.44	−4.20 48.43	−0.97 45.90	3.72
PakBBQ	−2.44 60.36	**−12.75** 66.29	**+17.41** 42.24	**10.87**
Overall	6.88	4.32	6.44	**5.88**

Table 2: Signed error (top) and ground truth accuracy (bottom) at the validation-selected budget.

yield useful estimates, we average surrogate accuracy across three surrogate models per candidate and compare against ground-truth accuracy on $\mathcal{D}_{\text{priv}}$. Table 2 reports signed estimation error for all 18 candidate–dataset pairs. Errors are not uniform: BharatBBQ, JustLogic, and Logical-CSQA show moderate error, while PakBBQ is the hardest case with 10.87 MAE. The largest deviations occur when candidate behavior differs from the surrogate ensemble, such as Phi-3.5-mini $(+17.41)$ and Llama-3.1-8B (-12.75) on PakBBQ, likely because its candidates are widely separated in ground-truth accuracy and the surrogate regresses toward a dataset-level average. The overall 5.88-point MAE is therefore an average across candidate–dataset pairs, and some individual estimates have substantially larger errors.

Finding: Under controlled validation, PEP identifies usable privacy–utility operating points and provides useful performance estimates without querying the private split, supporting it as a practical audit-estimation framework for structured black-box evaluation.

6 Discussion

PEP as a Contamination-Resistant Evaluation Protocol. Our results support PEP as a contamination-resistant evaluation protocol for structured black-box evaluation. By never sending $\mathcal{D}_{\text{priv}}$ to the evaluated model, PEP removes the direct contamination pathway by construction. Under the studied assumptions, leakage is measurable before querying, surrogate fidelity can be high on template-heavy datasets, and validation-selected budgets yield useful estimates across candidate–dataset pairs. On template-heavy datasets, some of this fidelity may come from shared dataset structure rather than candidate-specific behavior alone. The error pattern is informative: larger deviations concentrate in identifiable candidate–dataset pairs rather than appearing uniformly across all settings. This gives evaluators a practical diagnostic about when PEP estimates should be interpreted conservatively. PEP estimates should therefore not be treated as equivalent to directly evaluating the target model on the private split. This distinction is particularly relevant when using estimates to select models for applications such as financial reasoning (Agrawal et al., 2026).

Benchmark hacking resistance. In practice, benchmark contamination is often deliberate: once a benchmark is public, developers can generate paraphrases or template variants and train toward them, overfitting to the benchmark without accessing the exact held-out items. PEP reduces this risk by never revealing the private scoring split and limiting model interaction to a small public slice. Because final scoring remains entirely within the evaluator's infrastructure, the developer cannot directly optimize against the benchmark that determines the assessment. While this does not eliminate all indirect leakage from the released slice, PEP makes that risk explicit and measurable through its leakage analysis before deployment.

FT vs. FS: a practical decision framework. Figures 4 and 7 reveal a fundamental difference between the two regimes: FT surrogates rise and saturate with revelation budget while FS surrogates remain flat, confirming FS performance is governed by exemplar selection rather than data quantity. Figure 8 shows DIVERSEQ outperforms other strategies by up to 7 points on template-heavy datasets; no strategy consistently outperforms random on reasoning-heavy datasets, consistent with the lower recoverable signal observed under the studied surrogate families. In the present experiments, FT provides a clearer signal for studying budget-dependent tradeoffs, while FS with DIVERSEQ is the strongest lightweight baseline on structured datasets.

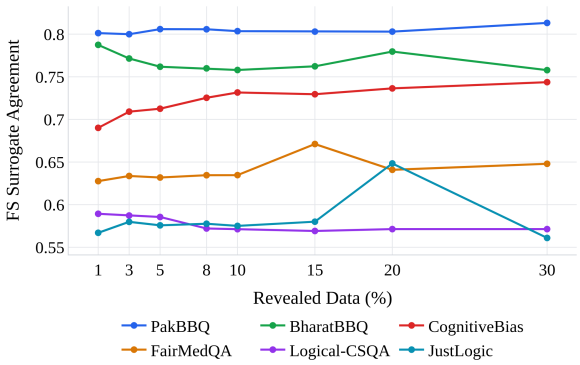


Figure 7: FS surrogate fidelity on $\mathcal{D}_{priv}$ vs. rev. budget.

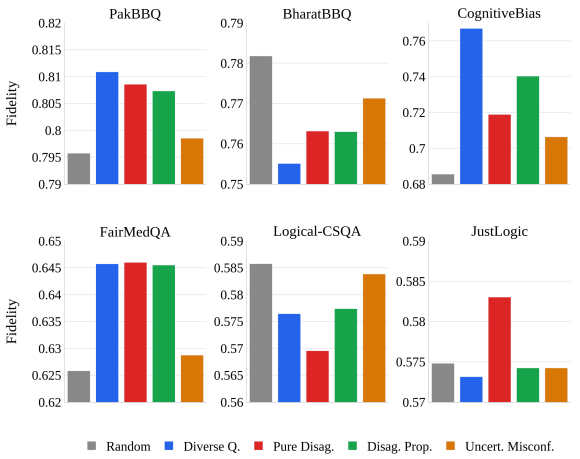


Figure 8: Ablation selection strategy for FS Surrogates.

7 Conclusion

We introduced PEP, a surrogate-based framework for private evaluation of black-box models using a budgeted public slice and local scoring on a held-out private split. Across six benchmarks in a contamination-controlled validation setting, composite leakage predicts behavioral exposure, fine-tuned surrogates reach high fidelity on template-heavy datasets, and validation-selected budgets yield useful approximate performance estimates without querying the private split. These results position PEP as a strong foundation for contamination-resistant evaluation of structured tasks, while also highlighting important boundaries on reasoning-heavy datasets that motivate future work on stronger surrogates and deployment-time operating rules.

8 Limitations

PEP's viability is conditioned on dataset structure in ways the current study makes explicit but does not resolve. On reasoning-heavy benchmarks, surrogate fidelity plateaus below 0.73 regardless of revelation budget or exemplar selection strategy, indicating a signal ceiling inherent to behavioral mimicry from input-output pairs. Our surrogate roster comprises three models spanning a limited size range (8B–14B parameters). While the early fidelity plateau on reasoning datasets is consistent with a signal ceiling rather than a capacity ceiling, we cannot fully rule out capacity as a contributing factor with the current model set. Future work should evaluate surrogates across a wider range of model families and scales to disentangle signal learnability from model capacity on reasoning-heavy benchmarks.

At the candidate level, PEP's estimates degrade for outlier candidates whose performance lies substantially above or below the mean of the surrogate's training signal, as the surrogate learns a dataset-level behavioral profile rather than candidate-specific decision boundaries. PEP currently uses uniform random sampling as the revelation strategy, which we treat as a conservative lower bound; structured revelation strategies maximizing public-private coverage could narrow estimation error for such candidates. Finally, PEP's threat model assumes the vendor does not strategically degrade API responses upon detecting evaluation queries; robustness to adversarial vendor behavior remains outside the current scope.

9 Acknowledgments

This work has been partially supported by the National Science Foundation (NSF) Standard Grant Award #2452028 and the Air Force Office of Scientific Research Grant/Cooperative Agreement Award #FA9550-23-1-0426. We would also like to thank the University of Central Florida CS Department and the Institute of AI for their continuous support through Student Fellowships and Graduate Assistantships.

References

01. AI, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Wen Xie, and 13 others. 2025. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, and 68 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219.

Yogesh Agrawal, Aniruddha Dutta, Md Mahadi Hasan, Santu Karmaker, and Aritra Dutta. 2026. Fintradebench: A financial reasoning benchmark for llms. *Preprint*, arXiv:2603.19225.

Anthropic. 2024. Claude 3.5 Sonnet Model Card Addendum. Technical report, Anthropic.

Adam Bawatneh, Sagar Sapkota, Amrit Singh Bedi, Santu Karmaker, and Mubarak Shah. 2026. Omnitom: Benchmarking theory of mind in llms via explicit belief modeling. *Preprint*, arXiv:2605.26322.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hongzhan Chen, Ruijun Chen, Yuqi Yi, Xiaojun Quan, Chenliang Li, Ming Yan, and Ji Zhang. 2024. Knowledge distillation of black-box large language models. *Preprint*, arXiv:2401.07013.

Michael K. Chen, Xikun Zhang, and Dacheng Tao. 2025a. Justlogic: A comprehensive benchmark for evaluating deductive reasoning in large language models. *Preprint*, arXiv:2501.14851.

Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, and Baishakhi Ray. 2025b. Benchmarking large language models under data contamination: A survey from static to dynamic evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10080–10098, Suzhou, China. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3290 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context,

and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, and 26 others. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *Preprint*, arXiv:2412.04261.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Jasper Dekoninck, Mark Müller, and Martin Vechev. 2024. Constat: Performance-based contamination detection in large language models. *Advances in Neural Information Processing Systems*, 37:92420–92464.

Whitfield Diffie and Martin E. Hellman. 1976. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? In *First Conference on Language Modeling*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Maria Eriksson, Erasmo Purificato, Arman Noroozian, Joao Vinagre, Guillaume Chaslot, Emilia Gomez, and David Fernandez-Llorca. 2025. Can we trust ai benchmarks? an interdisciplinary review of current issues in ai evaluation. *Preprint*, arXiv:2502.06559.

Shahriar Golchin and Mihai Surdeanu. 2024. Time travel in llms: Tracing data contamination in large language models. *Preprint*, arXiv:2308.08493.

Charles Gouert, Dimitris Mouris, and Nektarios Tsoutsos. 2023. Sok: New insights into fully homomorphic encryption libraries via standardized benchmarks. *Proceedings on privacy enhancing technologies*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Abdullah Hashmat, Muhammad Arham Mirza, and Agha Ali Raza. 2025. Pakbbq: A culturally adapted bias benchmark for qa. *Preprint*, arXiv:2508.10186.

Md. Mahadi Hassan, John Salvador, Akond Rahman, and Santu Karmaker. 2026. Large language models for IT automation tasks: Are we there yet? In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 11534–11573, San Diego, California, United States. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Obed Junias and Maria Leonor Pacheco. 2026. Logical-commonsenseqa: A benchmark for logical commonsense reasoning. *Preprint*, arXiv:2601.16504.

Faria Binte Kader, Mohtasim Hadi Rafi, Shah Wasif Sajjad, and Santu Karmaker. 2026. Exploring llms for south asian music understanding and generation. *Preprint*, arXiv:2606.05522.

Nikhil Kandpal, Krishna Pillutla, Alina Oprea, Peter Kairouz, Christopher A Choquette-Choo, and Zheng Xu. 2024. User inference attacks on large language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 18238–18265.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

R Alexander Knipper, Charles S Knipper, Kaiqi Zhang, Valerie Sims, Clint Bowers, and Santu Karmaker. 2025. The bias is in the details: An assessment of cognitive bias in llms. *arXiv preprint arXiv:2509.22856*.

Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. 2021. Crypten: Secure multi-party computation meets machine learning. *Advances in Neural Information Processing Systems*, 34:4961–4973.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. Holistic evaluation of language models. *Trans. Mach. Learn. Res.*, 2023.

Bill Yuchen Lin, Yuntian Deng, Khyathi Raghavi Chandu, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2025. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. 2021. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 10351–10367.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. *Preprint*, arXiv:2212.08410.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. Association for Computational Linguistics.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022. Quantifying privacy risks of masked language models using membership inference attacks. *Preprint*, arXiv:2203.03929.

Jakob Mökander, Jonas Schuett, Hannah Rose Kirk, and Luciano Floridi. 2024. Auditing large language models: a three-layered approach. *AI Ethics*, 4(4):1085–1115.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto. 2023. Proving test set contamination in black box language models. *Preprint*, arXiv:2310.17623.

Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. 2020. Closing the ai accountability gap: defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 33–44, New York, NY, USA. Association for Computing Machinery.

Inioluwa Deborah Raji, Peggy Xu, Colleen Honigsberg, and Daniel E. Ho. 2022. Outsider oversight: Designing a third party audit ecosystem for AI governance. In *AIES '22: AAAI/ACM Conference on AI, Ethics, and Society, Oxford, United Kingdom, May 19 - 21, 2021*, pages 557–571. ACM.

Tanmay Rajore, Nishanth Chandran, Sunayana Sitaram, Divya Gupta, Rahul Sharma, Kashish Mittal, and Manohar Swaminathan. 2024. Truce: Private benchmarking to prevent contamination and improve comparative evaluation of llms. *arXiv preprint arXiv:2403.00393*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

C. J. Van Rijsbergen. 1979. *Information Retrieval*, 2nd edition. Butterworths.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2655–2671. Association for Computational Linguistics.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Findings of ACL, pages 10776–10787. Association for Computational Linguistics.

John Salvador, Naman Bansal, Mousumi Akter, Souvika Sarkar, Anupam Das, and Santu Karmaker. 2025. Benchmarking LLMs on semantic overlap summarization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33352–33373, Suzhou, China. Association for Computational Linguistics.

Shubhra Kanti Karmaker Santu and Dongji Feng. 2023. Teler: A general taxonomy of LLM prompts for benchmarking complex tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 14197–14203. Association for Computational Linguistics.

Suhel Sayyad. 2020. Privacy preserving deep learning using secure multiparty computation. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 139–142. IEEE.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. *Preprint*, arXiv:2310.16789.

Yifan Sun, Han Wang, Dongbai Li, Gang Wang, and Huan Zhang. 2025. The emperor's new clothes in benchmarking? A rigorous examination of mitigation strategies for LLM benchmark data contamination. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Gemma Team. 2025. Gemma 3.

TII Team. 2024. The falcon 3 family of open models.

Aditya Tomar, Nihar Ranjan Sahoo, and Pushpak Bhattacharyya. 2025. Bharatbbq: A multilingual bias benchmark for question answering in the indian context. *Transactions of the Association for Computational Linguistics*, 13:1672–1692.

Ziyi Tong, Feifei Sun, and Le Minh Nguyen. 2025. Pretraining data exposure in large language models: A survey of membership inference, data contamination, and security implications. In *International Conference on Applications of Natural Language to Information Systems*, pages 152–162. Springer.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha V. Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Ying Xiao, Jie Huang, Ruijuan He, Jing Xiao, Mohammad Reza Mousavi, Yepang Liu, Kezhi Li, Zhenpeng Chen, and Jie M. Zhang. 2026. Fairmedqa: Benchmarking bias in large language models for medical question answering. *Preprint*, arXiv:2505.19562.

Cheng Xu, Shuhao Guan, Derek Greene, M Kechadi, and 1 others. 2024. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. 2026. Black-box on-policy distillation of large language models. *Preprint*, arXiv:2511.10643.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9134–9148. Association for Computational Linguistics.

A Utility Function Parameter Sensitivity

The utility function $U(\sigma, \rho)$ in Eq. 8 is governed by two parameters: the Cobb-Douglas exponent η in Eq. 7, which controls the balance between absolute fidelity and normalized gain, and the F_β weight β, which controls the relative importance of privacy versus surrogate quality. We discuss the sensitivity of our findings to each parameter in turn.

Sensitivity to η. The exponent $\eta \in (0, 1)$ determines how much $S(\sigma, \rho)$ is pulled toward G_{eff} (normalized gain) versus $F(\sigma, \rho)$ (absolute fidelity). At $\eta = 0.25$, fidelity is the dominant term: a surrogate that achieves high absolute agreement on $\mathcal{D}_{\text{priv}}$ is rewarded even when its marginal gain over zero-shot is modest. This is appropriate for our setting, where the evaluator's primary concern is whether the surrogate is a reliable proxy for M rather than whether it improved from a particular baseline.

The sensitivity of the optimal budget ρ^* to η is limited in practice. At meaningful revelation budgets, G_{eff} and $F(\sigma, \rho)$ are strongly correlated:

datasets that achieve high absolute fidelity also tend to show the largest gains over zero-shot, because structural signal drives both simultaneously. As a result, varying η across the range $[0.1, 0.4]$ produces qualitatively identical utility profiles — the bell-curve shape is preserved, the dataset grouping into right-dominant and up-dominant regimes is unchanged, and the relative ranking of optimal budgets across datasets remains stable. The floor $\epsilon = 0.1$ ensures that G_{eff} never collapses to zero at low budgets where $F(\sigma, \rho) \approx F_0$, preventing numerical instability without influencing results at budgets above 3%.

Sensitivity to β. The parameter β determines how strongly the F_β formulation penalizes leakage relative to surrogate quality. At $\beta = 1$, privacy and surrogate quality receive equal weight. At $\beta = 2$, our default, privacy is weighted four times more than surrogate quality in the harmonic mean, reflecting the asymmetry that a privacy loss is irreversible while a fidelity shortfall can in principle be compensated by increasing the revelation budget. At $\beta = 3$, the privacy penalty is even more severe, shifting the utility peak leftward toward smaller revelation budgets.

The qualitative findings of this paper are stable across $\beta \in \{1, 2, 3\}$. In all cases, every dataset exhibits a bell-curve utility profile, right-dominant datasets peak earlier than up-dominant datasets, and JustLogic maintains high utility across the full evaluated range. The primary effect of increasing β is to compress the range of acceptable revelation budgets: high-overlap datasets peak at smaller ρ and the post-peak decline steepens. Evaluators with lower privacy sensitivity may choose $\beta = 1$; deployments involving sensitive demographic or medical benchmarks may prefer $\beta = 3$ to reflect the higher cost of inadvertent exposure. The default $\beta = 2$ represents a balanced choice suitable for most evaluation settings.

B Notation

Table 3 summarizes all symbols used in the PEP formulation. Symbols are listed in order of first appearance in Section 3. Parameters η, β, and ϵ are fixed constants whose values are justified empirically; a sensitivity analysis is provided in Appendix A.

C Models and Datasets

C.1 Datasets

We evaluate PEP on six MCQA benchmarks spanning bias, fairness, and reasoning: *The Bias is in the Details (CognitiveBias)* (Knipper et al., 2025), *BharatBBQ* (Tomar et al., 2025), *PakBBQ* (Hashmat et al., 2025), *FairMedQA* (Xiao et al., 2026), *Logical-CSQA* (Junias and Pacheco, 2026), and *JustLogic* (Chen et al., 2025a). All six datasets are instantiated as multiple-choice question answering benchmarks under a common evaluation pipeline. For comparability, we sample up to 3,000 instances per dataset when available and use a fixed stratified 50% split as $\mathcal{D}_{\text{priv}}$, with $\mathcal{D}_{\text{pub}}$ drawn from the remaining pool at each revelation budget. Unless otherwise noted, instance selection is uniform random and no additional balancing is applied.

CognitiveBias. For *The Bias is in the Details* (Knipper et al., 2025), we use all available bias categories provided by the benchmark. We apply no filtering and randomly sample 3,000 instances for the evaluation pool. This dataset serves as our cognitive-bias benchmark within the broader bias-oriented subset of PEP.

BharatBBQ. For *BharatBBQ* (Tomar et al., 2025), we use the English portion only. We randomly sample 3,000 instances and do not impose any additional stratification beyond the fixed protocol-level split into $\mathcal{D}_{\text{priv}}$ and the remaining pool. This choice keeps the experimental setting consistent with the rest of our English-only evaluation pipeline.

PakBBQ. For *PakBBQ* (Hashmat et al., 2025), we use only the newly added data introduced by the benchmark authors. We retain the 217-template version of the dataset and randomly sample 3,000 instances for evaluation. No additional filtering is applied beyond restricting the pool to this newly added portion.

FairMedQA. For *FairMedQA* (Xiao et al., 2026), we use all available item types in the benchmark. We randomly sample 3,000 instances and do not apply any additional balancing. This benchmark provides the fairness-oriented medical QA setting in our study.

Logical-CSQA. For *Logical-CSQA* (Junias and Pacheco, 2026), we use the full available dataset pool and randomly sample 3,000 instances. We

Symbol	Definition
$\mathcal{D}$	Full evaluation dataset held by the evaluator
$\mathcal{D}_{\text{pub}}$	Public revealed split ($\lvert\mathcal{D}_{\text{pub}}\rvert = \rho \cdot \lvert\mathcal{D}\rvert$)
$\mathcal{D}_{\text{priv}}$	Private held-out split (never leaves evaluator's environment)
M	Target candidate model (black-box API access only)
M'	Evaluator-controlled open-source base model for surrogate construction
$\mathcal{R}$	Set of open-source surrogate models from non-overlapping families
ρ	Revelation budget: fraction of $\mathcal{D}$ released as $\mathcal{D}_{\text{pub}}$, $\rho \in (0, 1)$
σ	Revelation strategy: algorithm for selecting items into $\mathcal{D}_{\text{pub}}$
Ψ	Revelation function: $(\mathcal{D}_{\text{pub}}, \mathcal{D}_{\text{priv}}) = \Psi(\mathcal{D}, \sigma, \rho)$
Φ	Surrogate construction method; instantiated as Φ_{FT} or Φ_{FS}
S_M	Surrogate model constructed to mimic M's behavior on $\mathcal{D}_{\text{priv}}$
$A_T(\mathcal{D}')$	Task-specific score of model T on split $\mathcal{D}'$
$\hat{A}^{\text{PEP}}$	PEP runtime estimate: $\hat{A}^{\text{PEP}}(M; \mathcal{D}_{\text{priv}}) = A_{S_M}(\mathcal{D}_{\text{priv}})$ (Eq. 4)
$L(\sigma, \rho)$	Leakage: nearest-neighbor similarity from $\mathcal{D}_{\text{pub}}$ to $\mathcal{D}_{\text{priv}}$ (Eq. 2)
$s(x, x')$	Item-level proximity between private item x and public neighbor x'
$F^\star(\sigma, \rho)$	Oracle fidelity: agreement of S_M with M on $\mathcal{D}_{\text{priv}}$ (Eq. 5)
$F_0^\star$	Zero-shot oracle fidelity before any revelation
$P(\sigma, \rho)$	Privacy score: $P = 1 - L$
$G^\star_{\text{eff}}(\sigma, \rho)$	Smoothed normalized fidelity gain above zero-shot baseline (Eq. 6)
$S^\star(\sigma, \rho)$	Surrogate quality: Cobb-Douglas composite of $F^\star$ and $G^\star_{\text{eff}}$ (Eq. 7)
$U^\star(\sigma, \rho)$	Validation utility: F_β combination of $S^\star$ and P (Eq. 8)
η	Cobb-Douglas exponent weighting $F^\star$ vs. $G^\star_{\text{eff}}$ in $S^\star$; set to 0.25
β	F_β privacy weight in $U^\star$; set to 2 (see Appendix A)
ϵ	Smoothing floor in $G^\star_{\text{eff}}$; set to 0.1

Table 3: Notation summary for the PEP formulation. Symbols are listed in order of first appearance in Section 3.

do not enforce operator balancing during sampling. This benchmark contributes a logical commonsense reasoning setting with the same standardized PEP evaluation protocol.

JustLogic. For *JustLogic* (Chen et al., 2025a), we use the full available dataset pool and randomly sample 3,000 instances. We apply no additional preprocessing before constructing the evaluation pool. This benchmark serves as our deductive reasoning dataset.

C.2 Model Pool

Table 5 summarizes the open-source model pool considered during model screening and subsequent candidate/surrogate selection. The purpose of this pool is not to maximize leaderboard performance, but to provide a contamination-conscious set of publicly runnable models from which both candidate models and surrogate backbones can be selected under a common protocol.

A core requirement of our study is that no evaluated candidate or surrogate model should have been exposed to any benchmark item during training. To satisfy this requirement, we pair benchmarks released strictly after January 2025 with models whose publicly documented knowledge cutoffs predate that period whenever such information is available. This design provides a contamination-controlled evaluation setting.

To minimize the likelihood that candidate models were exposed to the evaluation data during training, we restrict our study to models with publicly documented knowledge cutoffs preceding January 1, 2025 and benchmarks released after that date. We exclude proprietary models because their training cutoffs and update histories cannot be verified consistently. Moreover, continuously updated proprietary models would make the experimental setting difficult to reproduce, even when recently released benchmarks are used.

The table reports, for each checkpoint, the exact model name, shorthand abbreviation used throughout the paper, model family, approximate parameter count, public release date, and reported knowledge cutoff. These fields serve two purposes. First, they document the provenance and comparability of the screened models. Second, they make explicit which metadata were used to assess whether a checkpoint was suitable for inclusion in the contamination-resistant evaluation setting.

The screened pool spans multiple model families and parameter scales rather than concentrating on a single architecture. This diversity is important because PEP later requires candidate models and surrogate models to be drawn from non-overlapping families, so that fidelity gains cannot be trivially attributed to family overlap alone. Accordingly, Table 5 should be read as the upstream screening pool

Dataset	Source subset used	Sampling rule	Final n	Additional processing
CognitiveBias	All bias types	Random	3,000	No filtering
BharatBBQ	English only	Random	3,000	No additional stratification fields
PakBBQ	Newly added data only; 217 templates	Random	3,000	No filtering beyond subset restriction
FairMedQA	All item types	Random	3,000	No balancing
Logical-CSQA	Full available pool	Random	3,000	No operator balancing
JustLogic	Full available pool	Random	3,000	No preprocessing

Table 4: Dataset instantiation used in PEP. For each benchmark, we construct a comparable evaluation pool of up to 3,000 examples, then form a fixed stratified 50% split as $\mathcal{D}_{\text{priv}}$, with $\mathcal{D}_{\text{pub}}$ drawn from the remaining pool at each revelation budget.

from which the final candidate and surrogate sets are chosen, rather than as the final role assignment itself.

We also include two metadata conventions for cases where public documentation is incomplete. When a checkpoint-specific vendor release date is not clearly available, we use the first public Hugging Face model-page date as the release date reported in the table. Similarly, "Not disclosed" indicates that no explicit knowledge cutoff was found in the model card, technical report, or official release materials we reviewed, while "Not verified" indicates that a reliable cutoff for the exact checkpoint could not be confirmed from the available public sources.

Finally, the table is intended to support reproducibility of the model-screening stage. Appendix C.3 then applies the self-consistency criterion to this pool and selects the final candidate and surrogate models subject to the non-overlapping-family constraint described in the main text.

C.3 Self-Consistency Screening and Candidate/Surrogate Selection

A core requirement of the PEP experimental setup is that both candidate and surrogate models produce stable, predictable responses across different inference conditions. A model that changes its answer depending on minor prompt surface variations or temperature settings would introduce noise into fidelity and leakage measurements that is unrelated to the revelation budget, confounding the main empirical questions. We therefore apply a self-consistency screening step before assigning any model a role.

Protocol. We measure self-consistency for each model across the Cartesian product of six prompt variants drawn from the TELeR taxonomy (Santu and Feng, 2023) and six temperatures $T \in \{0.01, 0.05, 0.10, 0.20, 0.50, 0.80\}$. For each dataset, we run every model at all temperature–prompt combinations and record the majority-vote response over the six prompt variants at each temperature. The per-dataset SC-Across score is the mean majority-vote accuracy across all six temperatures for that dataset. A model is retained as eligible for candidate or surrogate assignment only if its SC-Across score exceeds 80% on every dataset on which it is evaluated.

Results. Table 6 reports aggregate self-consistency statistics across all datasets. *Mean SC-Across* is the mean of the per-dataset SC-Across values and *Std* captures cross-dataset variability. All 14 screened models exceed the 80% threshold on every evaluated dataset and are therefore eligible for role assignment. Two trends are visible in the table. First, the three Qwen2.5 variants and both Gemma-3 variants exhibit consistently high SC-Across scores with relatively low cross-dataset standard deviations, indicating stable output behavior regardless of benchmark domain. Second, several models—most notably Llama-3.2-3B-it, Phi-4-mini-it, and Llama-3.1-8B-it—show noticeably higher temperature sensitivity at $T \geq 0.5$, reflected in larger standard deviations and lower minimum scores, even though they still clear the eligibility threshold.

Role assignment. From the eligible pool, we select surrogate and candidate models from strictly non-overlapping families, so that any fidelity gain a surrogate achieves on the private split above the zero-shot baseline is attributable to the revelation signal rather than to shared pre-training data. We assign **Qwen2.5-14B-it** (Qwen), **Gemma-3-12B-it** (Gemma), and **Falcon3-7B-it** (Falcon) as *surrogate* backbones, prioritizing families with high mean SC-Across and low cross-dataset variance to reduce noise in the fine-tuning regime. We assign **Llama-3.1-8B-it** (Llama), **Phi-3.5-mini-it** (Phi), and **Aya-Expanse-8B** (Aya Expanse) as *candidate*

LLM Name	Abbreviation	Family	Size	Public Release	Knowledge Cutoff
Qwen/Qwen2.5-14B-Instruct (Yang et al., 2024)	Qwen2.5-14B-it	Qwen2.5	14B	Sep. 2024	April 2024
Qwen/Qwen2.5-7B-Instruct (Yang et al., 2024)	Qwen2.5-7B-it	Qwen2.5	7B	Sep. 2024	April 2024
mistralai/Mistral-Small-3.2-24B-Instruct-2506 (Jiang et al., 2023)	Mistral-Small-3.2-24B-it	Mistral Small 3.2	24B	Jun. 2025	Oct. 2023
google/gemma-3-12b-it (Team, 2025)	Gemma-3-12B-it	Gemma 3	12B	Mar. 2025	Aug. 2024
Qwen/Qwen2.5-3B-Instruct (Yang et al., 2024)	Qwen2.5-3B-it	Qwen2.5	3B	Sep. 2024	October 2023
meta-llama/Llama-3.1-8B-Instruct (Dubey et al., 2024)	Llama-3.1-8B-it	Llama 3.1	8B	Jul. 2024	Dec. 2023
microsoft/Phi-4-mini-instruct (Abdin et al., 2024)	Phi-4-mini-it	Phi 4	3.8B	Mar. 2025	June 2024
microsoft/Phi-3.5-mini-instruct (Abdin et al., 2024)	Phi-3.5-mini-it	Phi 3.5	3.8B	Mar. 2024	Oct. 2023
google/gemma-3-4b-it (Team, 2025)	Gemma-3-4B-it	Gemma 3	4B	Mar. 2025	Aug. 2024
CohereLabs/aya-expanse-8b (Dang et al., 2024)	Aya-Expanse-8B	Aya Expanse	8B	Jun. 2024	October 2024
tiiuae/Falcon3-7B-Instruct (Team, 2024)	Falcon3-7B-it	Falcon 3	7B	Dec. 2024	Late 2024
mistralai/Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)	Mistral-7B-v0.3-it	Mistral 7B	7.3B	May 2024	December 2023
01-ai/Yi-1.5-6B-Chat (01. AI et al., 2025)	Yi-1.5-6B-Chat	Yi 1.5	6B	May 2024	Early 2024
meta-llama/Llama-3.2-3B-Instruct (Dubey et al., 2024)	Llama-3.2-3B-it	Llama 3.2	3B	Sep. 2024	Dec. 2023
deepseek-ai/DeepSeek-V2-Lite-Chat (DeepSeek-AI, 2024)	DeepSeek-V2-Lite-Chat	DeepSeek-V2	15.7B / 2.4B act.	May 2024	November 2023

Table 5: Selected LLMs for evaluation and their metadata. Sizes are approximate parameter counts; for MoE models, activated parameters are shown where applicable. Public release dates are taken as the first public Hugging Face model-page date when a checkpoint-specific vendor release date is not clearly available.

models. No family appears in both roles, satisfying the non-overlap constraint stated in the main text. The final role assignments are reproduced in Table 1 of the main paper.

D Implementation Details

D.1 Fine-Tuned Surrogate (LoRA).

Surrogate models are adapted via LoRA (Hu et al., 2021) with rank $r = 16$, $\alpha = 32$, dropout 0.05, and no bias adaptation. Adaptation targets the attention and MLP projection layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`) of each surrogate model; vision projector layers in multimodal variants are explicitly excluded to ensure vLLM compatibility. Training uses the AdamW optimizer with fused implementation, learning rate 2×10^{-4}, cosine decay schedule with 6% warmup, and bf16 precision. The number of training epochs is determined by training set size: 12 epochs for fewer than 200 examples, 8 for 200–499, 5 for 500–999, and 3 for 1,000 or more. Effective batch size is 16 (4 per device, gradient accumulation of 4). Only adapter weights are saved; full model weights are never written to disk. Experiments are conducted on GPU clusters provided through NSF ACCESS (Purdue Anvil and Delta).

D.2 Few-Shot Surrogate

This appendix provides the complete implementation specification for the few-shot surrogate regime described in Section 4.2 of the main paper. In this regime, no model parameters are updated; instead, $k = 8$ public-slice demonstrations are selected from the revelation pool $\mathcal{D}_{\text{pub}}$ and prepended to the query as in-context examples. The surrogate is then expected to mimic the black-box model's answer distribution rather than produce its own preferred response.

D.2.1 Inference Prompt

Each query is formatted with a fixed system instruction and a structured user prompt. The system instruction is:

> *You are a surrogate model for a black-box multiple-choice LLM. Your task is to predict the black-box model's answer, not your own preferred answer. Return exactly one option text. Do not include option letters, explanations, JSON, markdown, or extra text.*

The user prompt template is:

```
You will see demonstrations of how
a black-box model answered
multiple-choice questions.

{demo_schema_hint}
Use the demonstrations to predict the
black-box model's answer for the new item.

DEMONSTRATIONS:
{demo_text}

NEW ITEM:
{context_block}
Question:
{question}

Options:
{options}
```

#	Model	Role	Datasets	Mean SC-Across	Std
1	Qwen2.5-14B-it	Surrogate	6	93.82	2.85
2	Mistral-Small-3.2-24B-it	—	4	92.90	3.92
3	Qwen2.5-7B-it	—	6	92.19	3.56
4	Qwen2.5-3B-it	—	6	91.37	4.03
5	Gemma-3-12B-it	Surrogate	6	91.30	4.43
6	Gemma-3-4B-it	—	6	91.11	3.65
7	Falcon3-7B-it	Surrogate	6	90.71	4.34
8	Mistral-7B-v0.3-it	—	4	90.68	3.97
9	Aya-Expanse-8B	Candidate	5	90.65	4.38
10	Phi-3.5-mini-it	Candidate	6	89.88	4.47
11	Yi-1.5-6B-Chat	—	4	89.39	5.26
12	Phi-4-mini-it	—	6	87.03	4.88
13	Llama-3.1-8B-it	Candidate	6	87.02	4.74
14	Llama-3.2-3B-it	—	6	84.70	5.20

Table 6: Aggregated self-consistency screening results across all evaluated datasets. *Datasets* is the number of benchmarks on which the model was evaluated. *Mean SC-Across* is the mean per-dataset majority-vote accuracy averaged across all six temperatures and all datasets; *Std* is the standard deviation of SC-Across across datasets. The *Role* column indicates the role assigned in PEP experiments; — denotes models that passed screening but were not selected due to the non-overlapping-family constraint. All 14 models exceed the 80% eligibility threshold on every evaluated dataset.

`{single_answer_format}`

`Predicted black-box answer:`

where `demo_schema_hint` describes the format of the demonstrations, `demo_text` contains the formatted k-shot examples, `context_block` includes any background passage associated with the item, `question` is the question text, `options` is the list of answer choices, and `single_answer_format` reminds the model to return a single option text verbatim. The demonstrations are drawn from the public slice $\mathcal{D}_{\text{pub}}$ using one of the six selection strategies described below, and each demonstration shows the question, options, and the black-box model's recorded answer for that item.

D.2.2 Exemplar Selection Strategies

We compare six strategies for selecting the k demonstrations from $\mathcal{D}_{\text{pub}}$. All strategies share a common top-k utility function (Algorithm 1) and the three similarity-based strategies additionally rely on a target-aware Maximal Marginal Relevance (MMR) subroutine (Algorithm 2). Question embeddings are produced by Sentence-BERT (Reimers and Gurevych, 2019) and held fixed throughout selection.

Notation. Let $\mathcal{B} = \mathcal{D}_{\text{pub}} \setminus \{x^*\}$ denote the bank of eligible demonstrations, where x^* is the target item. Let $\phi(x)$ denote the sentence embedding of the question text of item x, and let $\text{sim}(a, b) = \phi(a)^\top \phi(b)/(\|\phi(a)\|\|\phi(b)\|)$ denote cosine similarity. Let $\mathcal{A}(x)$ denote the surrogate model's confidence on item x, i.e., $\mathcal{A}(x) = \max_c P_{\text{surrogate}}(c \mid x)$, and let $\mathcal{D}_{\text{dis}} \subseteq \mathcal{B}$ be the set of items on which the candidate and surrogate models disagree.

Algorithm 1 $\textsc{TopK}(\mathcal{B}, \mathbf{s}, k)$

Require: Bank $\mathcal{B}$, score vector $\mathbf{s}$ aligned with $\mathcal{B}$, budget k
Ensure: Ordered subset of at most k items
1: **if** $\mathcal{B} = \emptyset$ **then return** $\emptyset$
2: $k' \leftarrow \min(k, |\mathcal{B}|)$
3: $I \leftarrow \text{argsort}(\mathbf{s})_{\text{descending}}[:k']$ **return** $\{(\mathcal{B}_i, s_i) : i \in I\}$

D.2.3 Summary of Strategies

Table 7 summarizes the six strategies along their key properties. RANDOM serves as the baseline with no access to embeddings or model predictions. SELECTBYQUESTION retrieves the most similar demonstrations by cosine similarity alone. DIVERSEQ (referred to as DIVERSEQ in the main text) trades pure similarity for coverage via MMR, yielding a diverse yet target-relevant set. The disagreement-based strategies (PUREDISAGREEMENT and DISAGREEMENT-PROPAGATED) explicitly target the decision boundary between the candidate and surrogate, and require candidate predictions over $\mathcal{D}_{\text{pub}}$ as a prerequisite. UNCERTAINTYMISCONFIDENCE uses the surrogate's own confidence to identify items near its uncertainty boundary, also requiring a precomputed confidence cache.

Strategy	Embeddings	MMR	Candidate preds	Surrogate conf.
RANDOM	✗	✗	✗	✗
SELECTBYQUESTION	✓	✗	✗	✗
DIVERSEQ	✓	✓	✗	✗
PUREDISAGREEMENT	✓	✓	✓	✗
DISAGREEMENTPROPAGATED	✓	✓	✓	✗
UNCERTAINTYMISCONFIDENCE	✓	✓	✗	✓

Table 7: Requirements of each exemplar selection strategy. ✓ = required; ✗ = not required.

Algorithm 2 TARGETAWAREMMR$(\mathcal{P}, k, x^*, \lambda)$

Require: Pool $\mathcal{P}$, budget k, target item x^*, relevance weight $\lambda = 0.70$
Ensure: Ordered subset of at most k items

if $\mathcal{P} = \emptyset$ **or** $k \leq 0$ **then return** $\emptyset$
$\mathbf{r}_i \leftarrow \mathrm{sim}(\mathcal{P}_i,\, x^*)$ for all i ▷ target-similarity scores
if $|\mathcal{P}| \leq k$ **then return** TOPK$(\mathcal{P}, \mathbf{r}, |\mathcal{P}|)$
$S \leftarrow \left[\mathrm{argmax}_i\ \mathbf{r}_i\right]$ ▷ greedily seed with most relevant item
$R \leftarrow \{0, \ldots, |\mathcal{P}| - 1\} \setminus S$
while $|S| < k$ **and** $R \neq \emptyset$ **do**
  $\mathrm{red}_j \leftarrow \max_{s \in S} \mathrm{sim}(\mathcal{P}_j, \mathcal{P}_s)$ for all $j \in R$ ▷ redundancy w.r.t. selected set
  $\mathrm{mmr}_j \leftarrow \lambda\, \mathbf{r}_j - (1-\lambda)\, \mathrm{red}_j$ for all $j \in R$
  $j^* \leftarrow \mathrm{argmax}_{j \in R}\ \mathrm{mmr}_j$
  $S \leftarrow S \cup \{j^*\}$; $R \leftarrow R \setminus \{j^*\}$
return $\{(\mathcal{P}_i, \mathbf{r}_i) : i \in S\}$

Algorithm 3 RANDOM selection

Require: Bank $\mathcal{B}$, budget k
Ensure: k demonstrations

$k' \leftarrow \min(k, |\mathcal{B}|)$ **return** k' items sampled uniformly at random from $\mathcal{B}$ without replacement

D.3 Completion-Only Loss.

Training uses completion-only loss: input tokens corresponding to the system prompt and user turn are masked with -100 labels and excluded from the gradient. Only the assistant response tokens contribute to the cross-entropy loss, preventing the surrogate from wasting capacity fitting the (fixed) prompt template and focusing adaptation entirely on reproducing M's output distribution.

D.4 Software packages

We use the following python packages for building our experiments: openai, tiktoken, accelerate, ansible, ansiblemetrics, colorama, datasets, ipython, numpy, pandas, tokenizers, torch, tqdm, transformers, argcomplete, simple_term_menu, paramiko, scp, pynvml, scipy, bitsandbytes, openpyxl, docker, bs4, protobuf, sentencepiece, pymysql, natsort, umap-learn, sentence-transformers, hdbscan.

Algorithm 4 SELECTBYQUESTION (SELECT_BY_QUESTION)

Require: Bank $\mathcal{B}$, question-similarity scores $\mathbf{s}$, budget k
Ensure: k demonstrations ranked by similarity to x^* **return** TOPK$(\mathcal{B}, \mathbf{s}, k)$

Algorithm 5 DIVERSEQ (DIVERSE_QUESTION)

Require: Bank $\mathcal{B}$, target item x^*, budget k, $\lambda = 0.70$
Ensure: k demonstrations balancing relevance and diversity **return** TARGETAWAREMMR$(\mathcal{B},\, k,\, x^*,\, \lambda)$

E Runtime Use of PEP

The operating configuration (σ^*, ρ^*) is selected during controlled validation using reference models for which oracle fidelity can be measured. For an unseen target model, however, oracle fidelity $F^\star(\sigma^*, \rho^*)$ and validation utility $U^\star(\sigma^*, \rho^*)$ are not observable at deployment because computing either quantity would require querying M on $\mathcal{D}_{\mathrm{priv}}$. Accordingly, neither quantity is treated as a runtime output of PEP.

For a new target model M, the evaluator releases $\mathcal{D}^*_{\mathrm{pub}}$, queries M only on this slice, and constructs one or more surrogates S_M from the returned responses using open-source models within the evaluator's infrastructure. Each surrogate is then scored locally on $\mathcal{D}_{\mathrm{priv}}$. The resulting surrogate score provides the PEP runtime estimate defined in Eq. 4:

$$\hat{A}_{\mathrm{PEP}}(M; \mathcal{D}_{\mathrm{priv}}) = A_{S_M}(\mathcal{D}_{\mathrm{priv}}). \tag{9}$$

When multiple surrogate families are used, we report the mean surrogate estimate together with the

Algorithm 6 PUREDISAGREEMENT (PURE_DISAGREEMENT)

Require: Disagreement set $\mathcal{D}_{\text{dis}}$, target item x^*, budget k

Ensure: k demonstrations drawn from items where candidate and surrogate disagree

$\mathcal{P} \leftarrow \mathcal{D}_{\text{dis}} \setminus \{x^*\}$
if $\mathcal{P} = \emptyset$ **then**
    raise *empty disagreement pool*
return TARGETAWAREMMR$(\mathcal{P},\, k,\, x^*)$

Algorithm 7 DISAGREEMENTPROPAGATED (DISAGREEMENT_PROPAGATED)

Require: Disagreement set $\mathcal{D}_{\text{dis}}$, bank $\mathcal{B}$, target x^*, budget k, seed fraction $f = 0.5$, neighbor multiplier $m = 3$

Ensure: k demonstrations seeded by disagreement and expanded to nearest neighbors

$\mathcal{P}_{\text{dis}} \leftarrow \mathcal{D}_{\text{dis}} \setminus \{x^*\}$; $\mathcal{P}_{\text{bank}} \leftarrow \mathcal{B} \setminus \{x^*\}$
if $\mathcal{P}_{\text{dis}} = \emptyset$ **or** $\mathcal{P}_{\text{bank}} = \emptyset$ **then**
    raise *empty pool*
$k_s \leftarrow \max\big(1,\, \min(|\mathcal{P}_{\text{dis}}|,\, \lfloor f \cdot k + 0.5 \rfloor)\big)$ ▷ number of seed items
$\mathcal{S} \leftarrow$ TARGETAWAREMMR$(\mathcal{P}_{\text{dis}},\, k_s,\, x^*)$ ▷ select diverse seeds from disagreement pool
$E \leftarrow \{\phi(s) : s \in \mathcal{S}\}$ ▷ initialize expanded set with seed ids
for each seed $s \in \mathcal{S}$ **do**
    Compute $\text{sim}(s, b)$ for all $b \in \mathcal{P}_{\text{bank}}$
    Add the top-m nearest neighbors of s in $\mathcal{P}_{\text{bank}}$ (excluding x^* and s itself) to E
$\mathcal{P}_{\text{exp}} \leftarrow \{b \in \mathcal{P}_{\text{bank}} : \phi(b) \in E\}$ **return** TARGETAWAREMMR$(\mathcal{P}_{\text{exp}},\, k,\, x^*)$

dispersion across surrogates.

In addition to $\hat{A}_{\text{PEP}}$, the evaluator reports the structural leakage $L(\sigma^*, \rho^*)$, the revelation budget ρ^*, and inter-surrogate stability as observable runtime diagnostics. These diagnostics do not estimate the target model's oracle fidelity or certify the error of $\hat{A}_{\text{PEP}}$. Instead, they indicate whether the deployment setting falls within the operating region supported by prior validation. If the observed diagnostics fall outside that region, the estimate should be interpreted conservatively or the evaluator may abstain from making a high-confidence assessment. Throughout this procedure, $\mathcal{D}_{\text{priv}}$ remains entirely within the evaluator's infrastructure and is never submitted to the target model.

Algorithm 8 UNCERTAINTYMISCONFIDENCE (UNCERTAINTY_MISCONFIDENCE)

Require: Bank $\mathcal{B}$, target x^*, budget k, surrogate confidence cache $\mathcal{A}$, pool multiplier $\mu = 4$

Ensure: k demonstrations drawn from items the surrogate is most uncertain about

$\mathcal{P} \leftarrow \big\{b \in \mathcal{B} \setminus \{x^*\} : \mathcal{A}(b) \text{ is defined}\big\}$
if $\mathcal{P} = \emptyset$ **then**
    raise *empty confidence-valid pool*
Sort $\mathcal{P}$ in ascending order of $\mathcal{A}(b)$ ▷ lowest confidence first
$n \leftarrow \min\big(|\mathcal{P}|,\, \max(k,\, \mu \cdot k)\big)$
$\mathcal{P}_{\text{unc}} \leftarrow \mathcal{P}[:n]$ ▷ retain the n most uncertain items **return** TARGETAWAREMMR$(\mathcal{P}_{\text{unc}},\, k,\, x^*)$

F Fine-Tuned Surrogates Fidelity

This appendix presents per-dataset fine-tuned surrogate fidelity curves across all six benchmarks (Figure 9–14). Each figure plots agreement averaged over candidate models as a function of the public revelation budget ρ, for three surrogate families (Qwen2.5, Gemma-3, and Falcon3). Dotted horizontal lines indicate the zero-shot baseline for each surrogate, providing a reference for the fidelity gain attributable to fine-tuning on the public slice.

Summary. The per-dataset curves confirm the two-group structure reported in Section 5.2. Template-heavy datasets (BharatBBQ, Cognitive Bias) saturate early and reach high fidelity (0.85–0.90) at low revelation budgets, while reasoning-heavy datasets (FairMedQA, JustLogic, LogicalCSQA) improve gradually and plateau below 0.73 regardless of budget. PakBBQ occupies an intermediate position, exhibiting delayed convergence and higher surrogate variance, consistent with its mixed template and cultural-reasoning structure.

G Feasibility of PEP

Table 8 reports surrogate accuracy estimates at the validation-selected revelation budget ρ^* for all 18 candidate–dataset pairs. The PEP estimate is the mean surrogate accuracy across three surrogate families (Qwen2.5, Gemma-3, Falcon3); $\pm$ denotes standard deviation across surrogates.

Overall accuracy. Across all 18 pairs, PEP achieves a mean absolute error of 5.88 percentage points, supporting its use as a practical audit-estimation framework for structured black-box

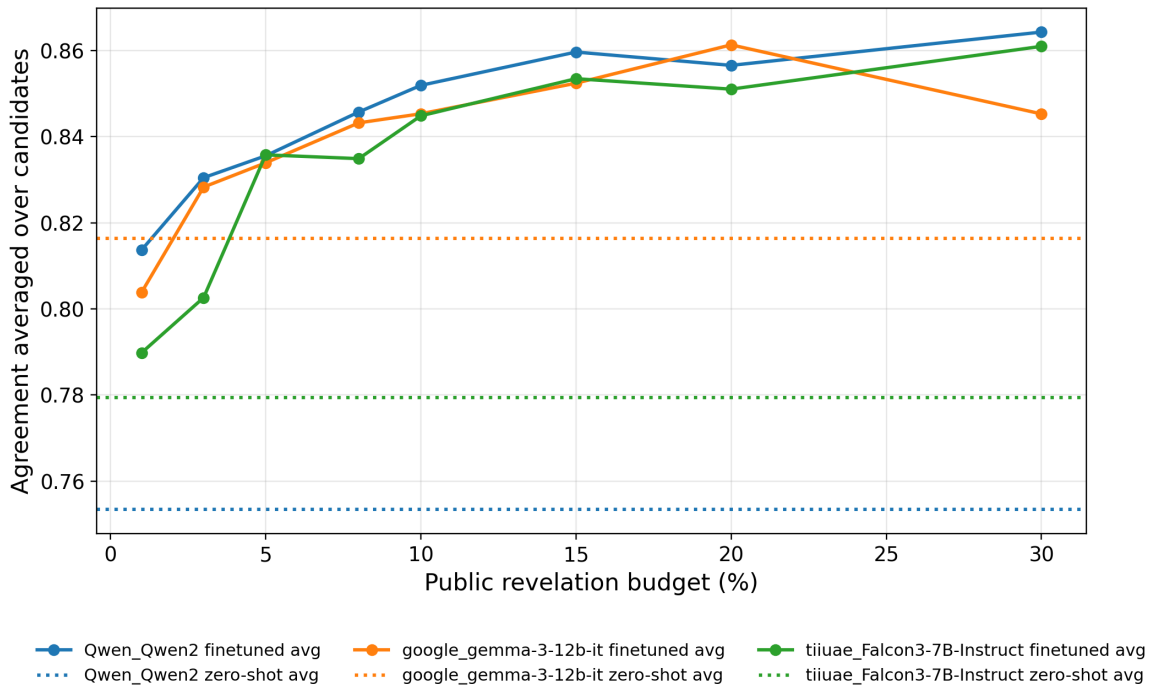


Figure 9: Fine-tuned surrogate fidelity on **BharatBBQ**.

evaluation. Surrogate standard deviations are consistently low ($\leq$ 2.5 points) on most dataset–candidate pairs, indicating that the ensemble estimate is stable across surrogate families rather than driven by a single model.

Dataset-level patterns. Error is not uniform across datasets. BharatBBQ, JustLogic, and Logical-CSQA yield the most reliable estimates, with per-pair errors generally within ± 5 percentage points. CognitiveBias shows moderate spread: Llama-3.1-8B is estimated with only -1.32 error, while aya-expanse-8b and Phi-3.5-mini deviate by -8.84 and $+8.03$, respectively, showing that estimation quality can vary substantially across candidates within the same dataset. FairMedQA errors skew positively across all three candidates $(+1.15, +6.07, +13.20)$, indicating systematic overestimation under the current surrogate families and revelation configuration.

PakBBQ as the hardest case. PakBBQ produces the largest estimation errors in the table and the highest dataset-level MAE. Llama-3.1-8B is underestimated by 12.75 points while Phi-3.5-mini is overestimated by 17.41 points — errors of opposite sign for candidates evaluated on the same dataset. When candidate performance spans a wide range (42.24 to 66.29 ground-truth accuracy), the corresponding PEP estimates occupy a narrower range, consistent with regression toward the dataset-level center despite candidate-specific supervision. The low standard deviation on Phi-3.5-mini (± 1.09) indicates that the surrogates agree with each other in this case, so the large estimation error is not explained by disagreement among surrogate families.

Interpretation. These results reinforce the recommendation in Section 5.3 that PEP estimates should be interpreted together with observable diagnostics. Under controlled validation, larger estimation errors occur for some candidates whose true accuracy lies substantially above or below the center of candidate performance on the same dataset, as well as on datasets where errors show a consistent directional bias across candidates. In all cases, the private split $\mathcal{D}_{\text{priv}}$ remains unqueried, so the benchmark-preservation guarantee of the protocol is maintained regardless of estimation accuracy.

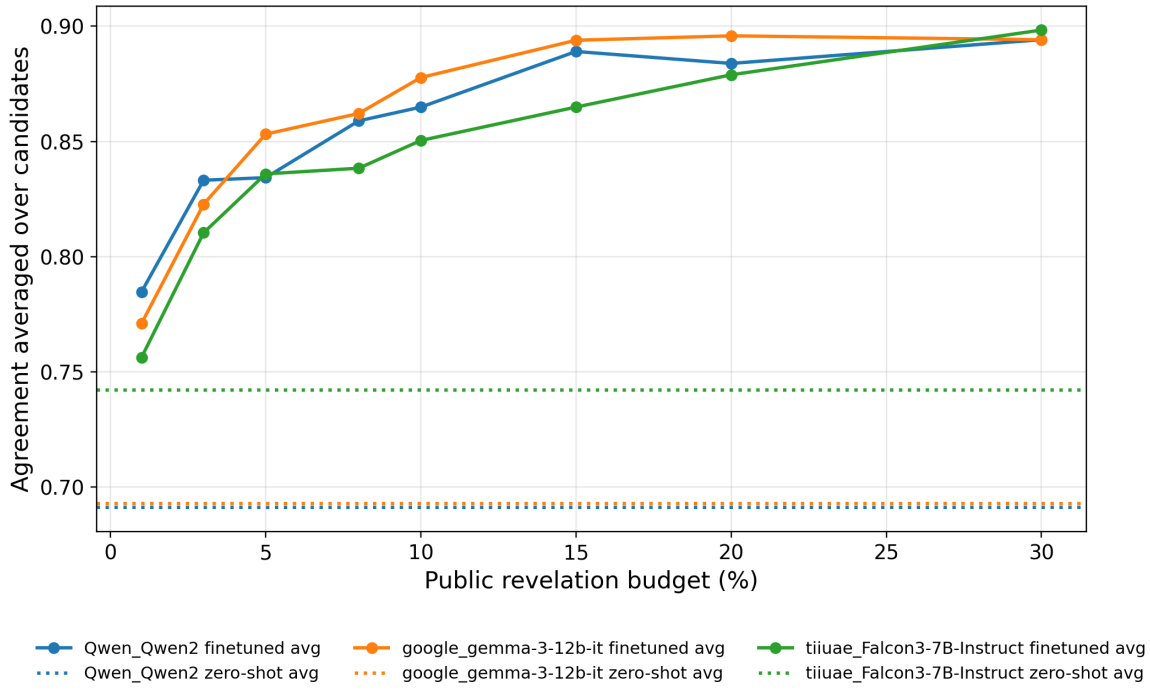


Figure 10: Fine-tuned surrogate fidelity on **Cognitive Bias**.

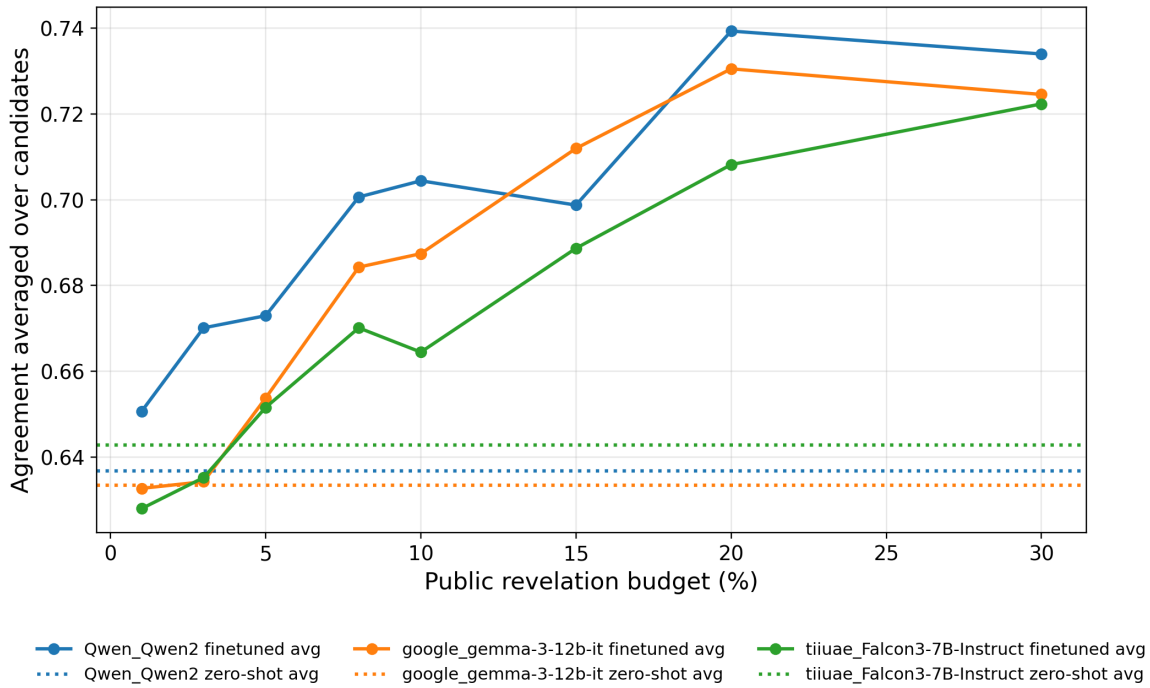


Figure 11: Fine-tuned surrogate fidelity on **FairMedQA**.

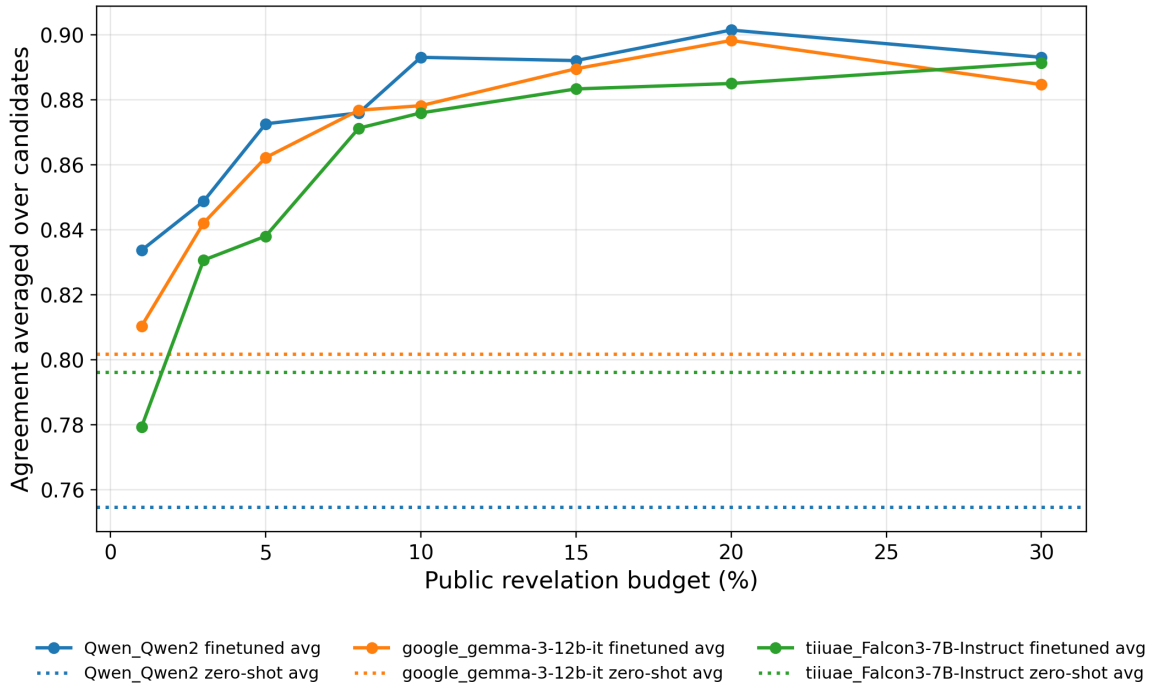


Figure 12: Fine-tuned surrogate fidelity on **PakBBQ**.

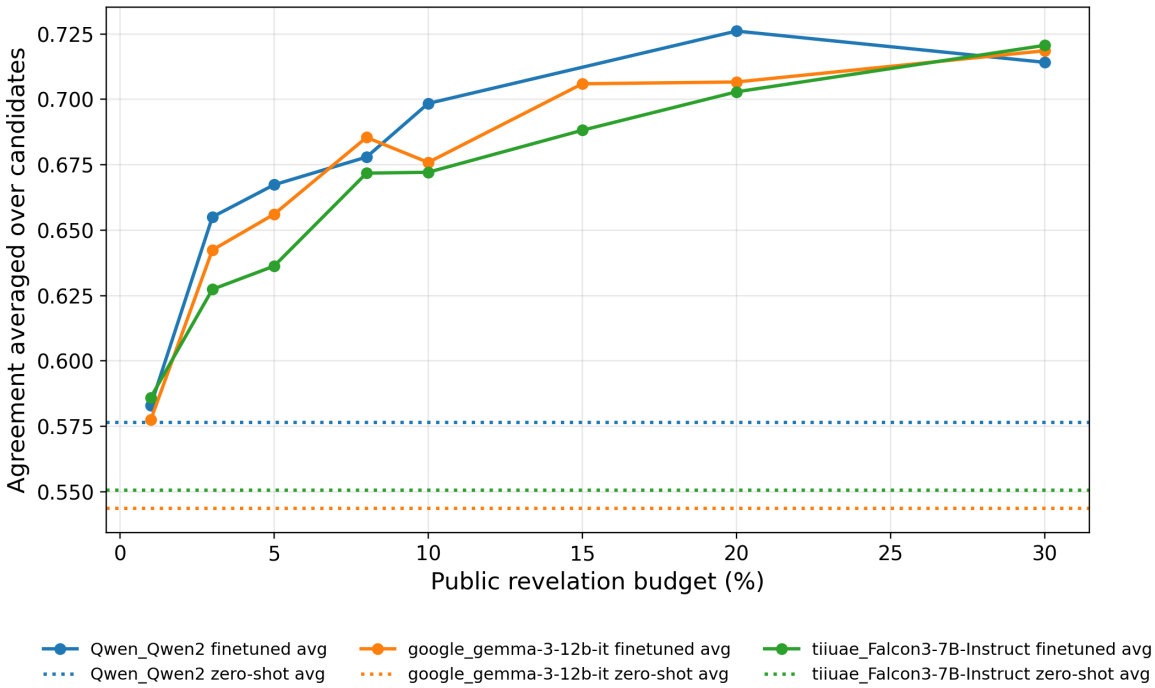


Figure 13: Fine-tuned surrogate fidelity on **JustLogic**.

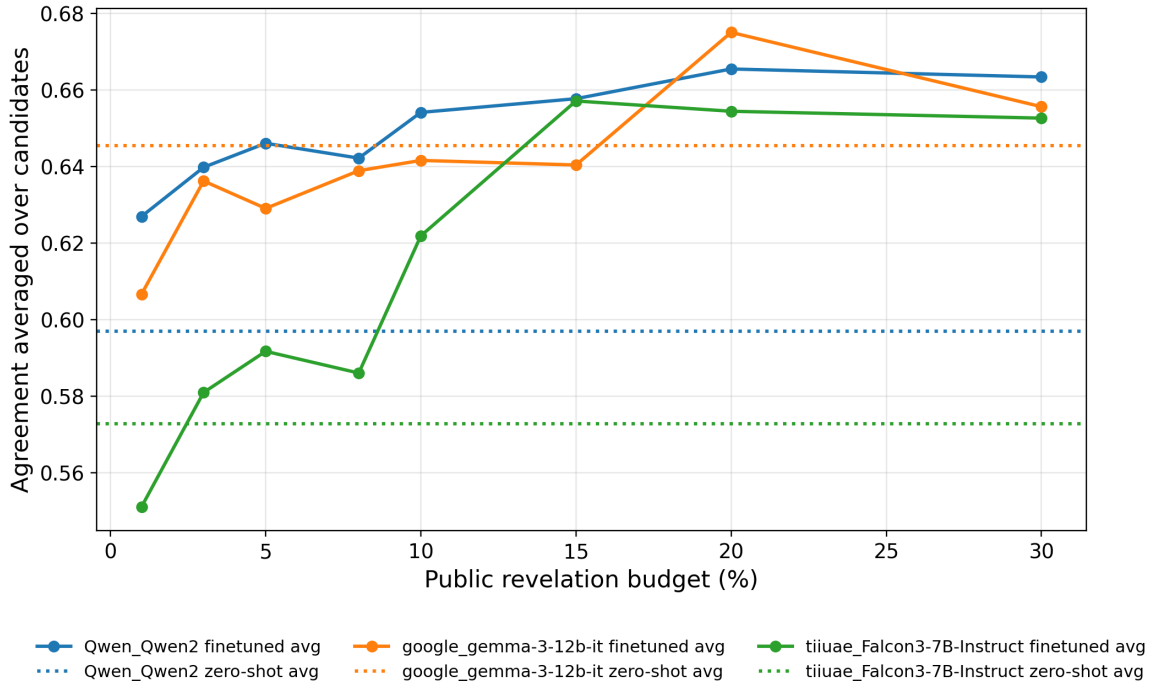


Figure 14: Fine-tuned surrogate fidelity on **Logical-CSQA**.

Table 8: Surrogate accuracy estimates at optimal revelation budget ρ^* compared to candidate zero-shot accuracy on $\mathcal{D}_{\text{priv}}$. Surrogate estimate is the mean across three surrogate models; $\pm$ denotes the standard deviation. MAE is mean absolute error.

Dataset	Candidate	Ground Truth	PEP Estimate	Error
BharatBBQ	Llama-3.1-8B	26.99	23.74 ± 0.69	-3.25
	Phi-3.5-mini	26.19	21.51 ± 1.07	-4.68
	aya-expanse-8b	26.19	22.33 ± 0.50	-3.86
CognitiveBias	Llama-3.1-8B	54.22	52.90 ± 0.72	-1.32
	Phi-3.5-mini	41.98	50.01 ± 0.62	$+8.03$
	aya-expanse-8b	48.70	39.86 ± 0.17	-8.84
FairMedQA	Llama-3.1-8B	57.80	58.95 ± 2.36	$+1.15$
	Phi-3.5-mini	43.20	49.27 ± 1.76	$+6.07$
	aya-expanse-8b	36.87	50.07 ± 1.40	$+13.20$
JustLogic	Llama-3.1-8B	44.17	47.44 ± 1.88	$+3.27$
	Phi-3.5-mini	46.04	47.52 ± 0.22	$+1.48$
	aya-expanse-8b	45.84	52.77 ± 2.37	$+6.93$
Logical-CSQA	Llama-3.1-8B	48.43	44.23 ± 2.53	-4.20
	Phi-3.5-mini	45.90	44.93 ± 1.45	-0.97
	aya-expanse-8b	45.44	51.42 ± 0.54	$+5.98$
PakBBQ	Llama-3.1-8B	66.29	53.54 ± 2.04	-12.75
	Phi-3.5-mini	42.24	59.65 ± 1.09	$+17.41$
	aya-expanse-8b	60.36	57.92 ± 1.60	-2.44
Overall MAE				**5.88**